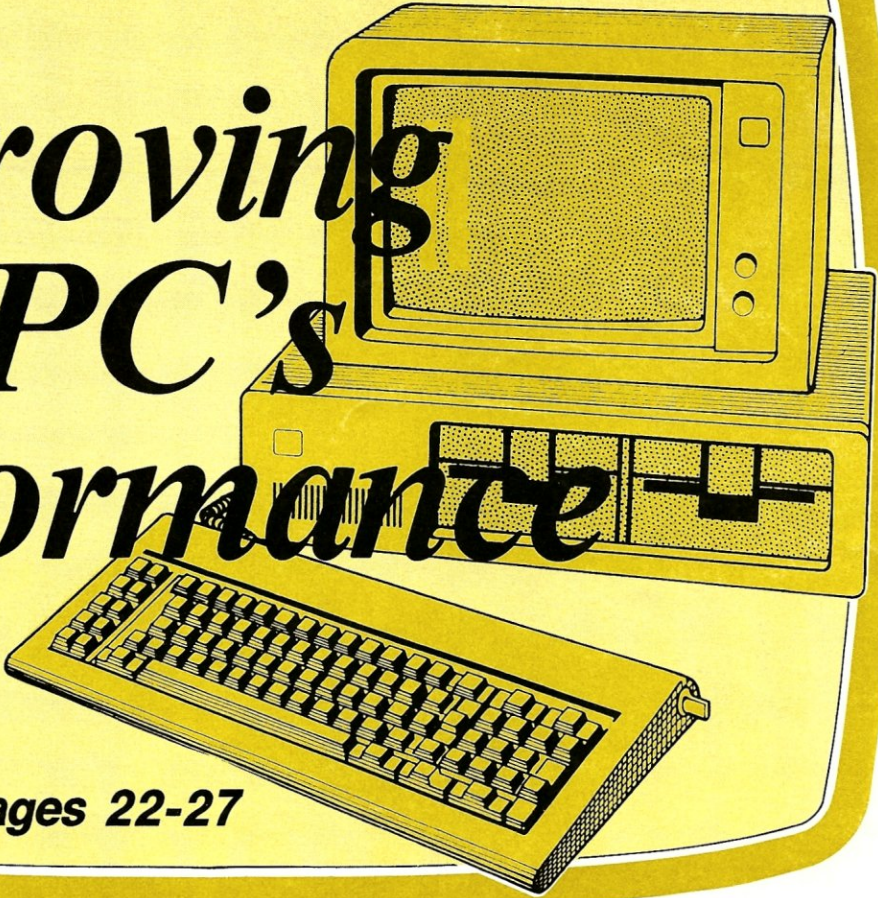


For the Advanced Computer User

Micro/Systems Journal™

Improving The PC's Performance

A yellow-toned illustration of a vintage personal computer setup. It includes a CRT monitor on top of a system unit, with a keyboard in front. The entire setup is enclosed within a yellow rounded rectangle. The text 'Improving The PC's Performance' is written in a large, stylized, italicized font across the middle of the illustration.

see pages 22-27

Also in this Issue

Creating A Copy Protected Program	28
Build A Smart Keyboard Interface	34
Program Interfacing To MS-DOS	54
Declare & Define C Variables In One File	66
More Loadable BIOS Drivers For CP/M	72

Complete Table of Contents on Page 3

July/August 1986

Vol.2/No.4

**\$4.50
U.S.A.**

MACROTECH—STILL THE S-100 PERFORMANCE PACESETTER

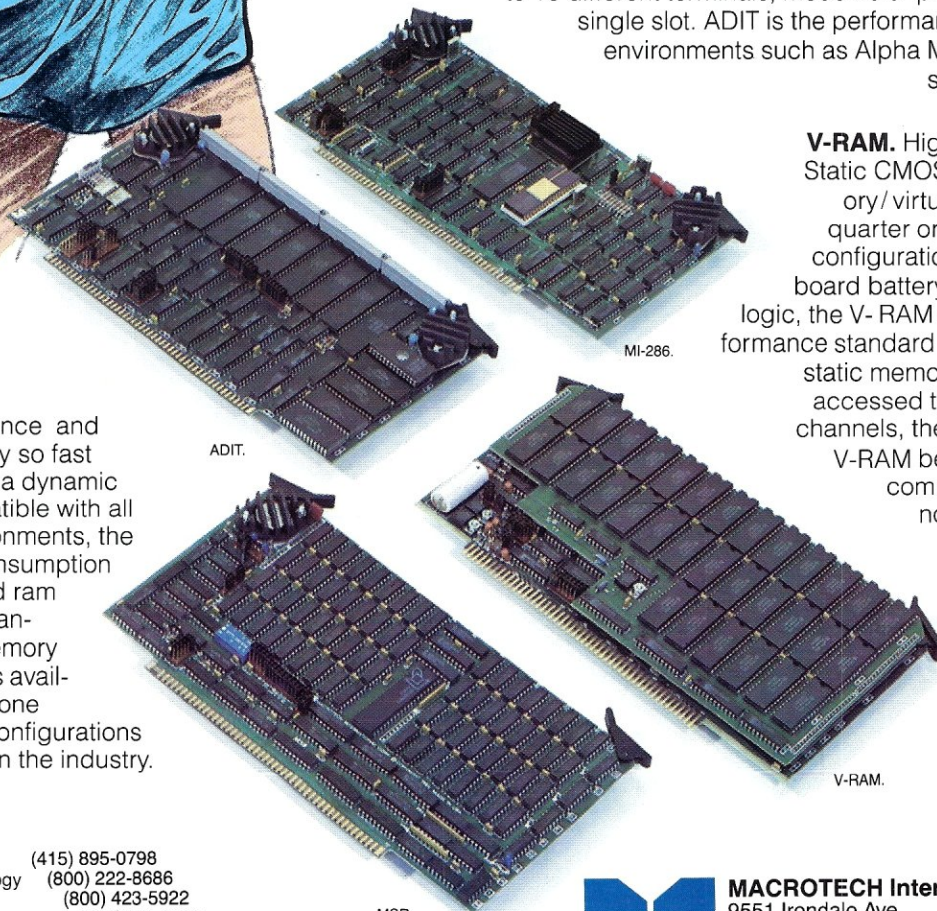


MI-286. Our 80286/Z80H Dual CPU Board is at least twice as fast as Compupro's 8085/88 and it's a direct replacement. The MI-286 has already become the standard by which other 80286 based systems are measured. Ask us for a complimentary Benchmark Report.

ADIT. There's nothing else like it on the market. It's an Intelligent I/O Board with its own real time firmware that lets you control up to 16 different terminals, modems or printers all from a single slot. ADIT is the performance standard in environments such as Alpha Micro where I/O speed is critical.

V-RAM. High performance Static CMOS system memory/virtual disk in either quarter or half megabyte configurations. With its on-board battery and power-fail logic, the V-RAM sets a new performance standard at conventional static memory prices. When accessed through I/O port channels, the half megabyte V-RAM becomes M Drive compatible with true non-volatile solid-state disk capability.

MSR. High performance and reliability in a memory so fast you won't believe it's a dynamic ram product. Compatible with all popular S-100 environments, the MSR's low power consumption and 120 nanosecond ram devices set a new standard for dynamic memory products. The MSR is available in quarter, half, one and two megabyte configurations at the lowest prices in the industry.



Dealers:

Gifford Computer Systems (415) 895-0798
Custom Computer Technology (800) 222-8686
S-100 (800) 423-5922
John D. Owens & Associates (212) 448-6298
In England; Fulcrum (Europe) Ltd. (0621) 828763

Macrotech dealers also include most Compupro Systems Centers, Heathkit Electronic Centers and Alpha Micro Dealers.



MACROTECH International Corp.
9551 Irondale Ave.
Chatsworth, CA 91311
(800) 824-3181 • in Calif. (818) 700-1501
Telex: 9109970653

MORE PERFORMANCE

In a multi-user system, performance is the most important ingredient.

Introducing the top-of-the-line in our 286 series, the **CompuPro 286/80™** with more performance built in than ever. For starters, our new 8 MHz 80286 processor gets data from memory in only two clock cycles, compared with the three or four required by other systems. In addition to the 768K of high speed memory, the system's high performance 80 Megabyte hard disk has a dedicated 512K byte cache buffer that boosts its performance even more. Tape back-up and floppy drives are included along with nine serial ports, Concurrent DOS 816™ multi-user multi-tasking operating system and NewWord™ word processing software. The system includes a 16 slot



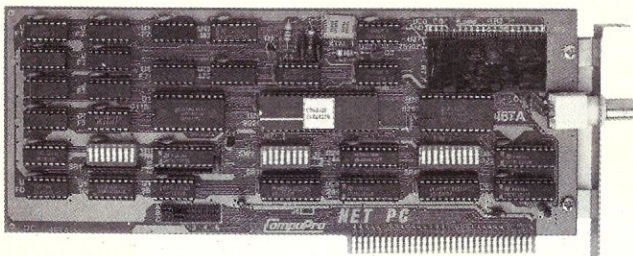
S-100R motherboard, which means there's plenty of room for adding options such as memory expansion to a megabyte, more cache memory, additional I/O ports, slave processors, graphics, networking and more.

All this tech-talk translates simply to this: The CompuPro 286/80 has the power to solve your business computing needs today and tomorrow as your business grows.

The price? A lot lower than you'd expect to pay for all this performance. Just \$12,500 suggested list. To find out how the CompuPro 286/80 (or any of our other multi-user systems) can give your business more performance, call today for the name of your nearest CompuPro dealer.

10 MHz available with **AMD 80286!** Dealers have details.

MORE NET PERFORMANCE



multi-user computers. Your problem of how to move up

TECH-TALK: All our network products use the high software protocols (except a PC with CPLINK, which emulates DOS 3.1 conventions). The network may contain up to 255 nodes. The NET PC is a fully compatible super-set of the SMC ARCNET card and therefore may be used with other network software systems.

NET PC with CPLINK: \$550. **NET 100** (for S-100 systems and the 286 series): \$495. **NET 10** (for the CompuPro 10 Plus™): \$495. All prices are suggested list.

CompuPro has been providing networking solutions for years with our **NET 100™** and **NET 10™** products, and we're proud to introduce the latest member of our network family — The **NET PC™**. Simply plug the NET PC into your IBM® PC, XT, AT or 100% compatible and your personal computer now becomes an integrated part of a high performance multi-user network. The **NET PC** includes the **CPLINK™** requestor program that lets your PC become a part of the network while still running DOS.

CompuPro multi-user systems (like the 286/80) make ideal file-servers for networked PCs, and they're still fully functional as to a true multi-user system and keep your existing PCs is now solved. speed (2.5 MBPS) ARCNET® token passing hardware protocol and DR NET™

AT COMPUPRO, PERFORMANCE IS NOT JUST ANOTHER BUZZWORD.

SUMMER HOT FLASH! Get a super summer deal on a CompuPro 10 Plus four user system with a 40 megabyte hard disk. For a limited time we're slashing the price to just \$5995! See your nearest dealer for more details.

CompuPro™

Viasyn Corporation, 26538 Danti Court
Hayward, CA 94545-3999 (415) 786-0909

CompuPro, 286/80, 10 Plus, NET 100, NET 10 and NET PC are trademarks of Viasyn Corporation. Concurrent DOS 816 is compound trademark of Digital Research, Inc. and Viasyn Corporation. DR NET is a trademark of Digital Research, Inc. NewWord is trademark of NewStar Software, Inc. CPLINK is a trademark of ComputerNet A/S. IBM is registered trademark of International Business Machines Corporation. ARCNET is a registered trademark of Datapoint Corporation. Prices and specifications subject to change without notice.



TURBONET®

One Strong Link Can Strengthen Your Whole System

TurboNET® is Teletek's new local area network that permits reliable high speed data transfers among computers of differing bus architectures. TurboNET is designed to be used in networks consisting of multiple S-100 based and PC based systems. It will allow up to 4000 users, including up to 255 IBM PCs or compatibles, to share a single network and all attached peripherals. The network can be organized in any number of different ways mixing Teletek's 8 and 16-bit multiuser systems and PCs in any combination.

Teletek's Networking Family consists of:

TurboNET PC: IBM-PC Network Interface Board

Teletek's TurboNET PC board offers IBM-PC Compatibility, CSMA industry standard protocols, 2 megabits/sec network speed, on-board CPU and communications management firmware, and media independence.

TurboNET S-100: S-100 Network Interface Board

Teletek's TurboNET S-100 board offers IEEE 696 Compatibility, CSMA industry standard protocols, 2 megabits/sec network speed, on-board CPU and communications management firmware, and media independence.

The benefits are obvious: The cost savings of shared peripherals, almost unlimited system expansion capability, and the use of existing PC workstations with the ability to run the myriad of application software written for MS-DOS and PC-DOS. This coupled with Teletek's 8 and 16-bit multiuser systems running application software written for CP/M and MP/M allows the system the ability to access almost any software library.

For more information on Teletek's TurboNET S-100 and TurboNET PC boards or on any of our full line of S-100 products, please call our Sales Department at 916-920-4600.



TELETEK

4600 Pell Drive, Sacramento, CA 95838
(916) 920-4600 Telex #4991834 Answer back — Teletek

© 1986 Teletek

For the Advanced Computer User

Micro/Systems Journal™

July/August 1986

Vol.2 No.4

STAFF

Publishers

Sol & Lennie Libes

Editors

Sol & Don Libes

Associate Editor

Susan Libes

Contributing Editors

A.G.W. Cameron

David Carroll

Ian Darwin

Nelson T. Dinerstein

Dave Hardy

Ken Jackson

Henry Kee

Steve Leon

William Wong

Editorial Assistant

Lenore Melillo

Advertising

Lennie Libes

Circulation/Administration

Lennie Libes

Linda Gohlke

SUBSCRIPTION RATES	1 Yr.	2 Yrs.
U.S.	\$20	\$35
Canada & Mexico	\$28	\$52
Europe & So. America	\$35	\$64
Asia, Africa, Australia & Near East	\$39	\$70

ADVERTISING RATES: Available on request. Call (201) 522-9347 or write to: M/SJ, Box 1192, Mountainside, NJ 07092.

CHANGE OF ADDRESS: Please send old label and new address to: Micro/Systems Journal, Box 1192, Mountainside, NJ 07092.

CORRESPONDENCE: Address all letters to: M/SJ, Box 1192, Mountainside, NJ 07092. The editor may also be contacted via MCI Mail (SLIBES).

Micro/Systems Journal (ISSN #8750-9482) is published bi-monthly by Libes Inc., 995 Chimney Ridge, Springfield NJ 07081. Second Class postage is paid at Springfield, NJ and at additional offices. POSTMASTER: send address changes to MICRO/SYSTEMS JOURNAL, PO Box 1192, Mountainside NJ 07092.

Copyright © Micro/Systems Journal, a subsidiary of Libes Inc. All rights reserved, reproduction prohibited without permission.

Micro/Systems Journal is a trademark of Libes, Inc.

IN THIS ISSUE

FEATURE ARTICLES

Power to the PC! 22

Stuart A. Jones

Creating A Copy Protected Program 28

Edwin Thall

Build A Smart Keyboard Interface 34

John Monahan

Program Interfacing To MS-DOS - Part VIII 54

William G. Wong

Declare & Define C Variables In One File 66

Bill Rogers

More Loadable BIOS Drivers For CP/M 72

Ted Carnevale

PRODUCT REVIEWS

Phoenix' PFI-X-Plus Debugger 48

Randy Davis

Holliston Challenger XT-186 52

Charles H. Strom

Magnum Digital PRO-180 & FD-100 62

Steven D. Kapplin

Hochstrasser's Modula-2 70

Edward J. Joyce

XPIP 84

Edward J. Joyce

DEPARTMENTS

Editor's Page 4

News, Views & Gossip 6

Sol Libes

The C Forum 8

Don Libes

Mail 10

The SIG/M Public Domain 14

Steve Leon

The PC/Blue Report 16

Hank Kee

The CP/M Bus 44

Robert A. Blum

The Unix File 68

Ian F. Darwin

The Scientific Computer User 80

A.G.W. Cameron

The Data Base Forum 88

Nelson T. Dinerstein

AUTHORS: Micro/Systems Journal is always seeking good articles. Please write or call first to see if we are interested in the subject. Please do not send the article unless we ask for it.

If you are interested in reviewing hardware or software please write telling us your interests, your background and include a sample of your writings.

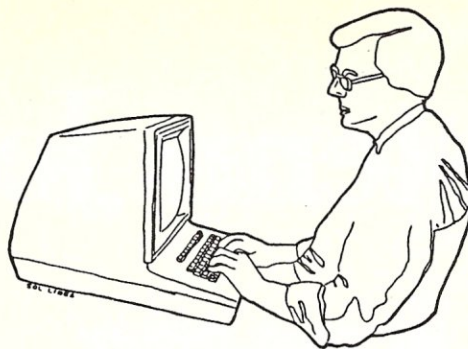
Send a stamped self-addressed business size envelope for a copy of our Author's Guide.

TRADEMARK ACKNOWLEDGEMENTS

Macintosh	Apple Computer Company
UNIX	AT&T
GEM, CP/M	Digital Research Inc.
Concurrent DOS	Digital Research Inc.
8088, 8086, 80286	Intel Corporation
IBM PC, PC/XT, AT	International Business Machines
MS-DOS, XENIX	Microsoft
GW Basic	Microsoft
68000, 68010, 68020	Motorola

Editor's Page

by Sol Libes



THE ORIGINS OF MS-DOS

It is interesting to look into the origins of MS-DOS. Microsoft gets the credit for creating what is, without a doubt, the most popular computer operating system in current use. But in actual truth Bill Gates, Microsoft's President, bought it from another company....Seattle Computer Products. And, if the truth be known, SCP never even wanted to create an operating system.

The story starts way back in 1979 when Intel introduced the 8086, the first 16-bit microprocessor. Seattle Computer Products, based in Seattle Washington, was a small start-up company with the sole goal of producing an 8086-based S-100 CPU card. They were very successful in designing the card and getting it into production. However, they had a problem. There was no software around for it.

Digital Research Inc. was working on porting CP/M over to run on 8086-based systems. However, CP/M-80 was going great guns for DRI and they were all involved with venture capitalists, acquiring other companies and laying the ground work for eventually going public and making a big killing. Thus, development of CP/M-86 was given a low priority status.

Microsoft also was involved with big OEMs, like Radio Shack, and did not have the time for small startups like SCP. SCP was successful in acquiring a license from Microsoft for MBASIC and hired a young programmer, Tim Paterson, to translate it into 8086 code so they would have at least one piece of software users could run on their SCP systems.

During 1980 SCP added a 16-bit wide memory card and disk controller cards to form a complete S-100 16-bit system. But they began to worry as they still did not have a disk operating system. DRI kept stringing SCP along, promising, but not delivering. Out of desperation, Tim Paterson quickly created an 8086-based operating system based on the CP/M-80 model. It had the same resident and transient functions and the same memory model. Before the year was out SCP was able to deliver complete systems to users.

DRI eventually released CP/M-86 in the Spring of 1981, just before IBM announced the PC.

ENTER IBM

In 1980, IBM realizing that Apple and Radio Shack were making inroads into their traditional market, the business office, decided they had to take quick action. They approached DRI and Microsoft with intent to license CP/M-80 and MBASIC, the personal computer industry standards at the time. They found Microsoft to be very receptive but DRI took a rather independent stance.

During one of IBM's visits to Microsoft they explained the problems they were having coming to terms with DRI. Bill Gates saw an opportunity. He sold IBM on the idea of using the new Intel 8088 microprocessor instead of the 8080 and billing the unit as a 16-bit computer, and that he would provide both a 16-bit version of MBASIC and a disk operating system.

But IBM was in a real rush to get the product out the door, before Apple and Radio Shack got too entrenched in the office market. Bill Gates, realizing that time was of the essence, decided to acquire SCP's operating system and do a quick translation of MBASIC.

Microsoft quickly stuck a deal with SCP. They bought the operating system for \$50,000, granting SCP a royalty-free license to MS-DOS when sold with their systems (including updated versions). Tim Paterson went to work for Microsoft.

IBM, with extensive experience with Disk Operating Systems and knowing their customer base, asked that the system be enhanced and made easier to use. But time was running out. So with only a few refinements, IBM released version 1.0 of PC-DOS. The biggest refinement was a user manual, written by IBM, that was designed for a less knowledgeable user and printed in a much more professional manner than traditional personal computer manuals. Anyone who had to cut their teeth on the CP/M-80 manuals realizes IBM must be given considerable credit for taking the industry a major step forward in documentation.

In early 1982 Microsoft delivered the improved version (2.0) of PC-DOS to IBM (it was released in mid '82). The new version represented a major improvement. By then IBM realized that users were going to

add hard disks to the PC and an operating system was needed that provided hard disk support. Thus, version 2 added many UNIX-like file handling functions, friendlier features (e.g. a powerful, easy to use batch processor) and error recovery. Version 2, in fact, bore little resemblance to SCP's original operating system or CP/M-80.

MICROSOFT GOES PUBLIC

Recently Microsoft went public and its chairman of the board, Bill Gates (age 30), sold 80,000 shares of stock for a little over \$1 million. He kept 11,142,000 shares (45% of the stock) worth over \$200 million. It is interesting to note that last year Bill's salary was \$133,000, while Jon Shirley's, Microsoft's President, was almost \$100,000 more than Bill's.

Paul Allen, co-founder of Microsoft with Bill (who has since left the company), owns 6.39 million shares worth well over \$100 million.

It is worth noting that Microsoft was built entirely out of income and did not rely on any venture capital. Sales for fiscal 1985 were over \$140 million (profits were over \$24 million) with well over half coming from OEMs. In fact, IBM, Tandy, and Compaq accounted for 31% and international OEMs (mainly Japanese) accounted for 19% of sales.

WHAT EVER HAPPENED TO SCP?

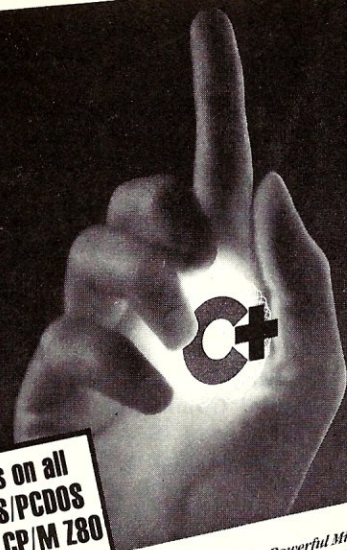
SCP continued to produce their S-100 8086 system throughout 1982, 1983, and 1984 with considerable success. They also began manufacturing PC-bus products. However, in late 1984 they experienced a plant fire which literally put them out of business.

Thus SCP decided, in early 1985, to sell the company, the most valuable asset of which was the royalty-free license to MS-DOS. In accordance with their contract with Microsoft they notified Microsoft of their intent to sell the company and assign the license to the purchaser. Microsoft, in turn, notified SCP that they could not do this.

The result is that SCP recently filed suit against Microsoft asking that Microsoft be enjoined from interfering with the sale and continuing with the royalty-free arrangement. Further, SCP is claiming that they are also entitled to any revenue from licenses sold to vendors that distribute systems based on chips other than the 8086. This would include IBM's entire line of personal computers. SCP is also seeking \$20 million in damages and if its other claims are not supported by the court, they are asking for a rescission of the original agreement with ownership of MS-DOS returned to SCP.

We will try to keep you posted on the outcome of this suit. (μ)

MIX C COMPILER



**Works on all
MSDOS/PCDOS
and all CP/M Z80
COMPUTERS!
(Not Copy
Protected!)**

The Powerful Mix
C COMPILER
*Harness the Power of the C Language
with this full featured Compiler*
MIX SOFTWARE

C FEVER *catch it!*

It's becoming an epidemic... everyone is switching to C! First there were a few hackers, then came the college students, next the major software houses, and now the rest of the programming world. Programmers everywhere are infected with the desire for SPEED, POWER, and PORTABILITY.

It's time to face the inevitable. You're going to catch the fever too! When you do, give us a call. We've got the best cure—an illustrated guide to the C language, plus a complete program development system. Everything you need to master the C programming language... all at a price that's less than the cost of a book!

But don't let this price fool you. Our system is powerful; it compiles twice as fast as the others, is completely standard, and it's very easy to use. Most C compilers are designed for wizards. We have designed ours for you!

What do you get for a mere \$39.95?

- A 450 Page book filled with sample programs, plus...
- A fast, standard, full featured C compiler that supports all data types and the latest features like bit fields, enumerations, structure assignment, and passing/returning structures.
- A fast linker that loads separately compiled files, searches libraries, and builds an executable program.
- An extensive library of more than 170 functions (including the standard C functions and the computer specific functions that provide direct access to the operating system and BIOS).
- Tools that allow you to optimize your programs for minimal space or maximum speed.

Incredible Value
\$39.95
AT ONLY With 30 Day
Money-Back Guarantee

Operators are standing by... Please use this Number for **ORDERS ONLY!**

CALL TOLL FREE FOR RUSH ORDER DELIVERY!

1-800-523-9520

**IN TEXAS, PLEASE
CALL TOLL FREE
1-800-622-4070**

For Technical Support Please call 1-214-783-6001

MIX Software, Inc. / 2116 E. Arapaho / Suite 363 / Richardson, Texas 75081

Or contact our Worldwide Distributors direct in:

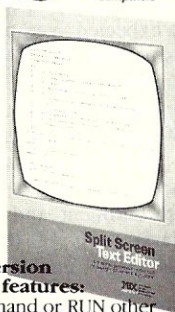
Canada: Saragway Software 1-416-923-1500 Switzerland: DMB Communication CH-1-825-53-29
Australia: Techflow 047-586924 France: Info/Tech 1-43-44-06-48

Split Screen Text Editor

an Incredible Value
\$29.95
AT ONLY

(Not Copy Protected!)
*Works on all MSDOS,
PCDOS and CP/M Z80
Computers*

Our high powered editor is great for editing high level languages. It works just like Micropro's Wordstar[®] but macros allow you to create your own custom editor, and the split-screen feature lets you edit two files at the same time.



The MSDOS/PCDOS version is loaded with special features:

- Execute any DOS command or RUN other programs from the editor.
- Quickly edit files as large as 300,000 characters.
- Compile MIX C programs directly from memory. The editor automatically positions the cursor to the first error in your program.

ASM UTILITY
an Incredible Value
\$10
AT ONLY

Call assembly language routines from your C programs. The ASM utility works with Microsoft's MASM or M80 assembler. Macros make it easy! Works just as if you were calling a C function, and you can even call C functions from assembly language. Lots of useful assembly language functions are included as examples.



RUSH REPLY ORDER FORM!

SPECIAL OFFER!

Limited Time Only
\$54.95
C Compiler & Text Editor

SAVE \$14.95 Off Our Regular List Price!

Please check method of payment:

☐ Check ☐ Money Order ☐ MasterCard/VISA

Your Card #: _____

Expires _____

Shipping Charges: (No charge for ASM Utility)

In the U.S.A.: Add \$5.00 per Order.

In CANADA: Add \$10.00 per Order.

OVERSEAS: Add \$10.00 per Text Editor. Add \$20.00 per C Compiler. Add \$30.00 for combined C Compiler and Text Editor.

Operating System: (Check one)

☐ CP/M Z80 ☐ MSDOS/PCDOS

Specify Your Computer Name _____

Specify Disk Format _____

NAME _____

Telephone A/C (_____) _____

Street _____

City _____

State _____

Country _____ ZIP _____

MIX software 2116 East Arapaho
Suite 363
Richardson, Texas, 75081
Ask about our Volume Discounts!
Call 1-214-783-6001 **M**

Description	Quantity	PRICE	Total Order
Split-Screen Text Editor	_____	\$29.95	\$_____
C Compiler	_____	\$39.95	\$_____
C and Text Editor (Special)	_____	\$54.95	\$_____
ASM Utility	_____	\$10.00	\$_____
Texas Residents Add 6.125% Sales TAX	_____	\$_____	\$_____
Shipping Charges (See at Right)	_____	\$_____	\$_____
TOTAL OF YOUR ORDER:	_____	\$_____	\$_____

News, Views & Gossip

by Sol Libes

GOSSIP & RUMORS

The Wall Street Journal recently described IBM as "the world's most profitable industrial company, it doubled in size in the past six years and could double again by the early 1990s." They also pointed out that "the company has rarely pioneered" but rather waited for smaller companies to create a market large enough to enter and then "bumped" them aside and grabbed "the largest share of the market."

The Wall Street Journal also stated that Dataquest, a respected market research outfit, "estimates that 80% of the 350 companies now supplying personal computers world-wide will go broke or give up the market by the end of 1987."

Quadram is rumored readying a \$600 (base price) AT compatible for introduction late this summer. The system will include 640K of RAM, power supply and keyboard, but no disk drives or controller.

Several dozen companies are rumored to already have 80386-based prototypes running and ready to go into production as soon as Intel starts delivering parts, expected to begin in late August. This means that we should see the first 386 machines being announced this fall. Compaq, AT&T, Tandy, and PC's Limited are expected to be front runners in the race to get systems out the door. IBM is not expected to release their 386 machine until next year and Microsoft is not expected to release its 386-based DOS and XENIX until early next year. This will, undoubtedly, cause some standards problems. The Far East manufacturers will likely wait until IBM enters the market before they jump in. Most 386 systems are expected to be designed for CAD, multi-user, network server, etc. applications, not single-user applications, and, early systems are expected to run Unix System V.

IBM and several OEMs are rumored beta testing Version 5 of PC/MS-DOS. This means that it should be released before year-end. It is expected to be specifically tailored for 286-based machines. The new version reportedly has more than 200 commands, offers protected mode operation, can address up to 16-Mbytes of memory, be multi-tasking, and use Unix-like shells. When released, it should have a disastrous effect on the sales of PC/XT machines.

IBM is also rumored evaluating a 1.44Mbyte (formatted), 3½" floppy drive

from Maxell Corp. Several drives makers are also known to be working on drives that will read both the current 720K and the upcoming 1.44Mbyte disks. It is expected that IBM will use the drive in a new version of the AT and in their 386-based system.

It is estimated that there are now over 500 companies making PC/AT bus products. That is more than S-100, Multibus, and VME buses put together. And one mail order outfit (Thompson, Harriman & Edwards) is offering a PC compatible for \$469 that includes 640K of memory and a single drive - adding a display controller, monochrome monitor and DOS brings an entry level system up to just under \$700. Some people are predicting that competition will force the price of a basic PC compatible down to \$500 by Christmas, but I doubt it.

Rumors are that Commodore shipped only 25,000 Amiga machines in its first six months of production, just a fraction of what they had expected to ship. Though the Amiga has been considered by many to be the most innovative machine introduced in 1985, most 68000 programmers have devoted their efforts to the Macintosh and Atari ST machines. Commodore slashed the price in the early spring so that a basic system, with color monitor, was offered, on sale, by some dealers for about \$1,000. This has served to stimulate sales, but how much? In addition to the lack of software, Commodore (as of press time) still was not shipping their long-promised IBM-PC emulator. An enhanced version of the Amiga, with a new operating system, is expected soon.

The Trenton Computer Festival offered some insights into industry trends. PC clones, with dual drives, and 640K of RAM, and mono display were going for under \$700. An XT clone was about \$1,000. Floppy disks were down to 45 cents each. And the Commodore Amiga, with color monitor, could be had for \$1,000.

Apple Computer has disclosed that it will offer versions of UNIX and MS-DOS operating systems for its new high-end 68020-based Macintosh, due in the late fall. Also, it disclosed that it now sells more Macintosh systems than Apple IIs. Does that mean that Mac sales are up, or that Apple-II sales are down?

Phoenix Software, Norwood MA, is offering OEMs the design of an under \$800

PC display controller offering 1280 x 1024-pixel resolution and five modes of operation.

Digital Equipment Corp. is rumored readying the "Vaxmate," an AT-compatible with built-in Ethernet circuitry.

19.2Kbaud MODEM INTRODUCED

I can still remember the old 110baud teletype days and the joys of moving to 300baud, 1200baud, and 2400baud modems. Last year saw the introduction of 9.6Kbaud and now we have the first 19.2Kbaud modem. It is a joint effort of Digital Communications Associates (Alpharetta GA) and Telebit Data Systems (Cupertino CA). The unit is actually an enhanced version of the DCA Fastlink unit (9.6Kbaud) and is usable on regular phone lines.

The new Fastlink unit also handles .3, 1.2, and 2.4Kbaud standard transmissions. It is available as a standalone unit (\$2,395) or PC-card (\$1,995). Owners of the 9.6Kbaud version of Fastlink can upgrade via a kit (\$99) or a factory upgrade (\$250). Either involves changing 4 ROMs.

INTERACTIVE CD ROMS

CD ROMs are the new emerging technology and developers are already talking about the next generation of systems....the interactive compact disk (ICD). The first ICD unit may appear before the end of the year.

ICD is an enhanced CD ROM with graphics and audio added to data storage. The ICD would interface directly to display and audio systems and include the Digital-to-Analog converters to deliver digitized video images and sound directly. The first units expected to reach the market are expected to employ a 68000 microprocessor, use the OS-9 operating system (currently used in the Radio Shack color computer among others), provide 768 x 580 pixels and stereo sound.

The first units to reach the market are expected to be designed strictly for entertainment and not contain computer interfaces. It will be interesting to see how these things develop.

UPGRADING THE AT TO 32-BITS

Intel has been demonstrating an AT with a small board that holds an 80386 and plugs into the 80286 socket, turning the AT into a limited 80386 machine. Several companies already have 386 products for the AT in beta test, so it's likely that the products will be released before year's-end. Two of the companies are ARIEL (201-788-2788) and BGI (215-538-3900).

We can also expect to see similar products based on the new NEC V-70 32-bit and V-60 16-bit processors. Both are expected to have enhancements such as built-in math processors and have 286

object code emulation modes. The V60 is already being sampled (\$550) and production is expected to start in August. The Intel/NEC court case should drag on through the courts for about two years depressing sales of the NEC V series chips. Thus, it is unlikely that U.S. manufacturers will adopt the V60/70 chips for use in their systems. NEC is claiming that the V60 is ten times better than the 68020 and 80386 chips. The V70, expected next year, has full 32-bit i/o.

LOW COST 32-BIT WORKSTATIONS RUMORED

More details on the new Commodore and Atari 32-bit workstations are leaking out. Commodore is rumored readying a system based on the Motorola 68020 for introduction in June with deliveries to start in the late fall. The system is expected to sport 1024x720 pixel resolution in monochrome and color with a base price under \$3,000. Included will be a 20Mbyte hard disk, SCSI interface port, 1 Mbyte of RAM (expandable to 4Mbytes and possibly more), five internal slots (proprietary bus), 3 1/2" drive (880Kbytes) and 14.3Mhz clock speed.

And Atari is expected to announce their 32-bit system at about the same time. Based on the Western Electric WE32000, it is expected to sport a 1024x1024 mono or color display.

Both systems are expected to run the UNIX operating system and are aimed at specialty markets such as university, engineering/scientific and video graphics producers.

A QUICK LOOK AT THE ATARI 1040ST

Atari introduced the 520ST in June of '85 and less than 9 months later introduced the 1040ST. With 1Mbyte of RAM, an internal 720K floppy and other enhancements, the machine appears to be intended to compete with the Commodore Amiga and Apple Macintosh. And there are rumors an empty IC socket inside the 1040ST is for a graphics coprocessor chip capable of 1000x1000 pixels. Couple this with the fact that Atari has purchased an AT&T UNIX license and there is reason to suspect Atari may enter the scientific workstation market with this system.

Atari is moving very quickly. Although the amount of available software is still meager by comparison to the Mac, it is better than what's available for the Amiga. There are already desktop publishing programs available which should challenge the Mac in the only market niche in which the Mac has achieved any success.

However, Jack Tramiel still appears to be up to his old tricks of demoing prototype products at shows and then not putting

them into production. Two recent examples are a promised low-cost CD ROM player and a box containing a 5 1/4" drive and the ability to run PC software on the 520ST and 1040ST.

FIRST WORM SYSTEM INTRODUCED

Information Storage Inc., 2768 Janitell Rd, Colorado Springs CO 80906, has introduced the first optical WORM (Write Once Read Mostly) drive for the PC (\$2,495). It store 115-Mbytes in a standard 5", full-height, form factor and is furnished complete with interface card and software. It uses a replaceable data cartridge (\$60). At this price, and considering that most of the software on a hard disk usually does not change, it presents a very attractive alternative to a second hard disk drive.

RANDOM BITS

Elliam Associates, 24000 Bessemer St, Woodland Hills CA 91367 (tel: 818-348-4278) has a new 100-page catalog listing over 8,000 CP/M public domain programs (on 400 disks). The catalog is \$7.50 (\$10 overseas) and disks are available in 75 different formats on 8", 5 1/4", and 3 1/2" disks....IBM recently presented a paper at the International Conference on Acoustics, Speech and Signal Processing describing a PC-based real-time voice recognition system capable of recognizing 20K words. In 1984, IBM had demoed a 5K word system requiring an entire IBM 4341 mainframe with three 190L floating point systems array processors....IBM got rid of its 84 computer stores by selling them to Nynex Corp. They have learned what DEC and Xerox and several hundred computer store operators have learned - selling personal computers via stores is a hard sell.

COMPUTER FLEA MARKETS

July 6: Wilkes-Barre PA. Ham & Computerfest, Coal St. Sports Complex, run by Murgas A.R.C. (KeYTL), (717)388-6863.

July 20: Cherry Hill NJ. Computer Fair, Holiday Inn, Cherry Hill NJ, Rt 70, run by Robert Barlow (201)533-1991.

August 3: Suffern NY. Computer Fair, Holiday Inn, run by Robert Barlow (201)533-1991.

August 16: Philadelphia PA. Lasalle University, 20th & Olney Ave., run by Phila. Area Computer Society, (215)951-1255.

August 23/24: Dayton OH. Computerfest-86, Hara Arena, run by Dayton Microcomputer Assoc. (513)268-7225.

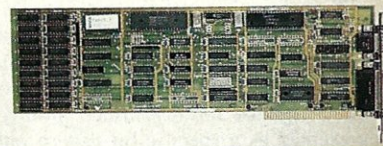
August 31: Lancaster PA. Red Rose Computerfest, Gurnsey Sales Pavilion, on Rt 30 East at Rt 896, run by Red Rose Repeater Assoc., Box 5029, Lancaster PA 17601. [M]

THE WORLD'S FASTEST Z-80 COMPUTER IS NOW A PC



EARTH COMPUTER's TURBOSLAVE-PC™ is the world's fastest Z-80 Coprocessor. Running at 8MHz, it was designed to permit operation of thousands of CP/M application programs on your IBM-PC, XT, AT™, or compatible computer system.

The **TURBOSLAVE-PC** supports the TurboDOS™ multi-user operating system which allows up to 16 users on your PC. It is the only IBM-PC/Z-80 system that is MP/M™ compatible and allows **TRUE** multi-user, multi-process operations, including full record locking and security.



Discover a whole new world of high-speed (8MHz) single and multi-user applications for your personal computer. Discover the **TURBOSLAVE-PC**... the world's fastest Z-80 Coprocessor, with such outstanding features as:

- 128K RAM with parity
- 2 Serial ports
- On-board Counter Timer
- S.L.R. Z-80 assembler included

To order your **TURBOSLAVE-PC**, call or write to:



EARTH COMPUTERS

P.O. Box 8067, Fountain Valley, CA 92728
TELEX: 910 997 6120 EARTH FV

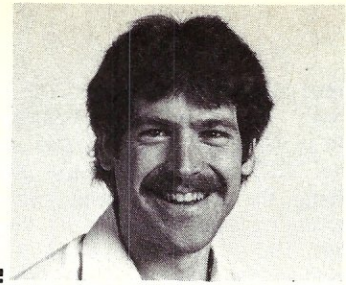
(714) 964-5784

Ask about **EARTH COMPUTERS'** other fine PC and S-100 compatible products.

IBM-PC, XT, AT are trademarks of International Business Machines, Inc.; CP/M and MP/M are trademarks of Digital Research; TurboDOS is a trademark of Software 2000; TURBOSLAVE-PC is a trademark of Earth Computers

The C Forum

by Don Libes



To advance from a novice to an experienced (but not necessarily expert) C programmer, one must be familiar with *casts* (also referred to as *coercions* or *conversions*). Grizzled C programmers know when to use them or avoid their use through good declarations. Beginning C programmers stumble into them, or sometimes avoid them through use of lucky declarations that may fail mysteriously when their program is ported to another computer.

Intuitively, a cast converts one type to another. The syntax of a cast is a type name in parentheses followed by the expression to be cast. For example, to produce a null character pointer, zero is cast as follows:

```
(char *)0
```

To paraphrase K&R (p.42), "a cast has the same effect as an assignment to a variable of the casted type." Internally, this means that a value's bit patterns may potentially change due to a cast. Some casts actually increase or decrease the number of bits used in representing the value.

If you don't want the bit patterns to change, you probably want to use a union. A practical example of this is the problem of comparing an IEEE-style floating point number against infinity. Since infinity is represented as a specific bit pattern rather than a number, there is no way to specify infinity in the natural notation of the type. One solution is to declare a union of a float and an integer long enough to store the infinity bit value, and then use the integer type when testing for infinity.

GENERAL GUIDELINES

If you find yourself always casting a variable before using it, you should probably re-examine your declaration, and ask yourself why it isn't declared as the casted type to begin with. In general, casts tend to disappear when variables are declared appropriately.

However, there are valid reasons for casting and I will discuss the major ones.

POINTER CONVERSIONS

The first one I want to get out of the way is the NULL problem.

```
#define NULL 0
```

```
char *broiled = NULL;
```

The C code above correctly assigns *broiled* to be a null pointer. This is because

Not Cast In Concrete

assignment implicitly converts the right-hand side to the type on the left-hand side. However, such conversions do not take place during function calls. Because of that, the following statement is incorrect.

```
function_takes_pointer_arg(NULL);
```

There are two problems with this statement. They stem from the fact that C does not check function calls against function definitions, so that this statement will have the compiler place an int zero on the stack.

The first problem is that the size of the pointer that the function expects may be different than the space required to hold an int. If the pointer is larger, extra bits accessed will have garbage values in them. If smaller, following arguments will not be accessed correctly. Redefining NULL as, say, (char *)0 or 0L, is not correct either since pointers of different types are not guaranteed to be the same size.

The second problem is that the NULL pointer may use a non-zero bit representation. Then, even if `sizeof(pointer) == sizeof(int)`, the values won't match, unless C does the conversion. The cast will do this conversion correctly. (Conversely, casting a null pointer to an int is defined to return the integer 0.)

If you remember only one thing from this column, let it be the following: *Passing NULL as a pointer argument ALWAYS requires a cast.*

Two functions that invariably require casts are `malloc()` and `free()`. `malloc()` returns space guaranteed to be alignable to any type, but is declared as returning (char *). Just as the results of `malloc()` are almost always cast, similarly the argument to `free()` should be cast back to (char *).

Because of `malloc()`'s guarantee, it is acceptable to recast it to any pointer type. However, recasting of arbitrary pointers may lose information, depending on a machine's memory alignment rules. Even worse is what happens if you explicitly violate memory alignment restrictions. I unthinkingly did this while writing a program on a Motorola 68000. I was reading

variably-sized objects out of a buffer that I was managing myself. Occasionally I would get a bus fault while reading a two-byte integer. The code looked something like:

```
char *bufptr = buffer;
```

```
2_byte_integer = *(int *)bufptr; /* ok */
bufptr += 17;
2_byte_integer = *(int *)bufptr; /* bus fault! */
```

The bus fault would occur whenever `bufptr` was odd. This condition was set up because there was no padding between elements in the buffer and some of the elements were an odd-number of bytes in length. The 68000 requires word loads and stores to be aligned on even-byte boundaries, and I was violating this, hence the bus fault.

Various things combined to mislead me, including 1) porting it to a VAX (which doesn't have word-store alignment restrictions) causing the problem to *disappear*; and 2) the debugger I was using cleverly faked the odd-address access and executed the request by doing two loads without telling me that!

Remember that when you cast a pointer, you are assuming responsibility for pointer compatibility.

FUNCTION CASTING

A special type of pointer casting is that of functions (which are treated as pointers). While function casting is rather unusual, a common enough cast is the following (extracted from the include file "signal.h" on all UNIX systems):

```
#define BADSIG (int (*)( ))-1
```

This definition of `BADSIG` coerces the type of `-1` to be a *pointer to a function returning an int*. It's hard to imagine what `-1` might do if called as a function. In fact, it is never actually called. `-1` is only cast that way so that it can be checked as a return value against `signal()`, which is defined to return, you guessed it, a pointer to a function returning an int. What `signal()` is telling you when it returns a `-1`, is that you have handed it a *bad signal*, hence the name `BADSIG`.

ARITHMETIC CONVERSIONS

More typical uses of casts appear in arithmetic expressions (which includes character manipulation). The most common reason is needing to use a library call like `sqrt()` which is defined for doubles, but you have an int. In that case, the following suffices.


```
double sqrt();
int x, y;
x = (int)sqrt( (double)y);
```

Here we have cast twice, first to promote `y` to the type required by `sqrt()`, and second to convert the double returned by `sqrt()` to an int. Notice that the second cast was, in effect, a truncation (or *demotion*) from a float to an int and accuracy was lost. Sometimes this is exactly what is wanted. Sometimes this is an error, since the double can store much larger numbers than the int.

CASTS TO AVOID SIGN PROPAGATION

Operations that propagate the sign bit can be rectified by casting from a signed to an unsigned type. For example, storing a signed character with the high bit on into an int variable will result in a negative value. Similarly, shifting signed ints can propagate sign bits.

In both cases, you can avoid this by either specifically declaring your variables as unsigned, or casting them as (unsigned ...) in expressions. Be especially aware, that some compilers treat chars as signed unless they are explicitly declared unsigned. Arbitrarily declaring all chars (and or any other types) unsigned is a mistake, however, because you may pay a significant performance penalty for using it instead of plain char.

CASTS TO AVOID NUMERIC OVERFLOW

It is occasionally useful to cast operands to larger types to avoid overflow. For example,

```
double z;
int x = MAXIMUM_INTEGER;
```

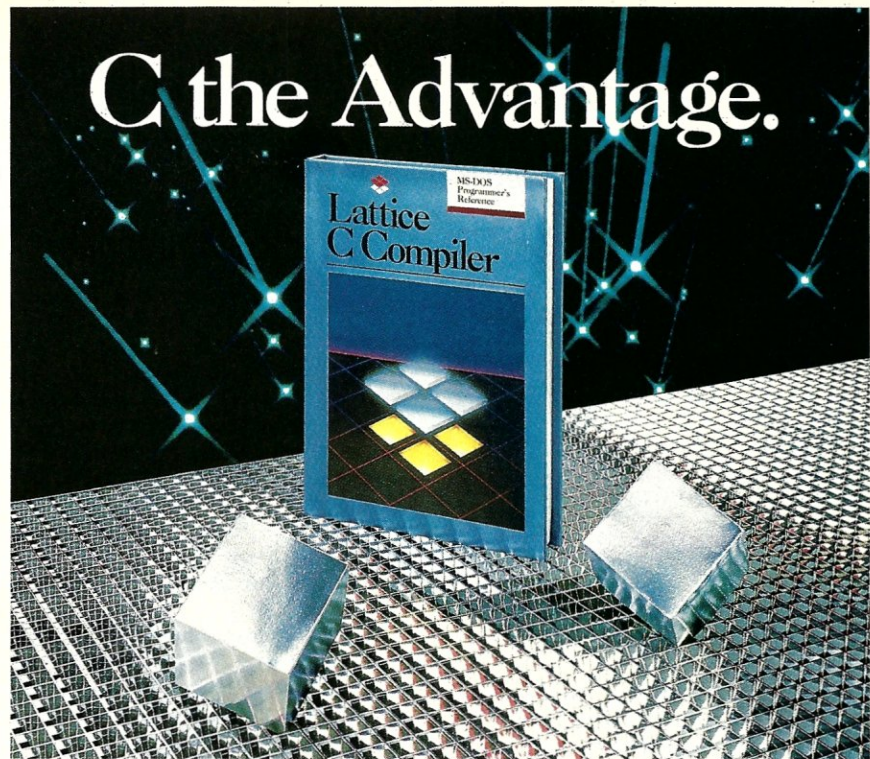
```
z = x + x; /* overflow */
z = x + (double)x; /* ok */
```

The first attempt at the addition fails because the addition of two ints is an int, which is not large enough for the result. The second attempt works because the addition of an int and a double is a double which is large enough for the result. Note, that this example doesn't depend on the result being assigned to a double. The result of the addition expression alone is the problem. You must always be careful to check whether your intermediate results need such a cast, even if you know that the final result fits.

DOCUMENTING A COERCION THAT WILL OCCUR ANYHOW

Occasionally, it is useful to use a cast as documentation that a conversion is going to take place anyway. This might not actually be clear otherwise.

In function calls, for example, chars are promoted to ints, and floats to doubles. Therefore, the following calls are equiv-



There's never been a better time to buy Lattice C. Professional programmers the world over have made Lattice C the standard compiler for serious MS-DOS programming. Our compiler features include: ANSI language constructs including, *unsigned* as a modifier, *void* data type, *enum* data type, structure assignments, structure arguments, structure returns, and argument type checking.

The library contains more than 200 new functions, including: ANSI/UNIX/XENIX compatibility; extended support for MS-DOS; extended support for networking including file sharing, file locking, and I/O redirection; and flexible error handling via user traps and exits. Plus the library has also been re-engineered to produce much smaller executables.

Try the new Lattice C Compiler. Because C-ing is believing.



Lattice, Incorporated
P.O. Box 3702
Glen Ellyn, IL 60138
312/858-7950 TWX 910-291-2190

INTERNATIONAL SALES OFFICES:

Benelux: Ines Datacom (32) 2-720-51-61 Japan: Lifeboat Inc. (03)293-4711
England: Roundhill (0672)54675 France: SFL (1)46-66-11-55
Germany: (49)7841/4500 (49)8946/13290

alent even though they don't appear to be if you aren't familiar with the automatic promotion of arguments in C function calls. (It was certainly a surprise to me, when I first heard it!)

```
float f = 522.9347;
char c = 'Z';
```

```
function(f,c);
function( (double)f,(int)c);
```

Certain other operations also cause automatic conversions, which are occasionally usefully pointed out by putting in redundant casts. These include returning values from functions, assignment, some miscellaneous arithmetic and logical operations.

CASTING TO VOID

Any value may be cast to (void). Normally, this is only used to state that you are discarding a return value of a function.

For example, if you believe the rule that *if you can't handle an error, you shouldn't be checking for it*, this might appear as,

```
(void) fprintf(stderr, "uh oh...");
```

After all, if this fails, what should the program do? (μ)

Don Libes is a computer scientist working in the Washington DC area. He works on artificial intelligence in robot control systems.

there is mail.....



We welcome your letters with comments, compliments, criticism and suggestions. We read them all and publish the most noteworthy, even if they are critical of us. We do not have the staff to answer all letters personally. And all letters become the property of M/SJ and may be subject to editing. Further, we do not print letters that do not include a name and address.

Please send your letters to: Micro/Systems Journal, Box 1192, Mountainside NJ 07092.

SOLVING THE DISPLAY MEMORY MYSTERY

Dear Sol,

I recently had quite a surprise which might interest you. I had written a PC screen intensive benchmark, which a V20 microprocessor should have been able to complete in roughly half the time of a standard 8088. Actual measurements were quite disappointing, showing little improvement (5.7 seconds for the V20 versus roughly 8 for the 8088). Confused, I went back to the instruction set specifications for both processors and counted clock cycles. I discovered two things: 1) more than 98% of the time was being spent in a single (gigantic) block move of screen memory which scrolled the screen from right to left, and 2) not only should the V20 have been twice as fast, but it should have completed the benchmark in roughly 2 seconds vs. 4 for the 8088.

Thinking that perhaps someone was fudging their instruction timings, it occurred to me that although the actual times were way off, the difference between the two was almost exactly the calculated 2 seconds (8 vs. 5.7). Maybe time was being lost somewhere. Although the program was designed to handle a graphics display, just as a lark, I redirected the program to the monochrome screen and retimed it. Amazingly I got a time of 4.2 seconds! I then redirected the program to a nonexistent screen at 8000:0, and measured 2.4 seconds, very close to the calculated times.

Could the display memory be holding the CPU off? A check of the IBM Tech Ref Manual showed that the motherboard has wait state hardware and that it is tied to a signal coming from the video cards called I/O CMPLT. Using the NEC figure of 8 clock cycles per byte of block move, I inferred from the timings that the CPU is being held off of screen memory an average of 6 clock cycles per access in the

monochrome, and 11 clock cycles per move in the graphics card. But a block move involves two memory accesses per cycle (a read followed by a write). Does that mean that our figures are 2 times too high for a single fetch? Rerunning a similar benchmark with the MOVs replaced with a STOS, which only performs a single memory fetch per cycle, results in wait states of 6 and 12, almost exactly the same.

What seems to be happening is that the video card only allows the CPU access to video memory for short periods of time: longer than it takes for a MOVs to make a single read and write cycle, but shorter than it takes for the MOVs or STOS to come around again with the next transfer. During this time the CPU has completely unhindered access to display memory (no wait states). It then holds the CPU completely off of memory for 6 clock cycles in the monochrome card and 12 in the C/G card (independent measurement indicate 10 cycles in the EGA), presumably while the CRT controller reads up the next block of pixels/characters to display. Turning the CRT controller chip off would probably result in the number of wait states going to zero for both cards, but that would result in screen flicker.

Stephan R. Davis

We trust that you will recognize the above letter writer's name. Randy has authored several articles in M/SJ, including the two-part series on the NEC V20 chip. Thanks for the info Randy! Keep the tidbits coming.

C-DEBUGGER CRITERIA

Dear Sol,

I think Jonathan Sachs missed the boat in his review of C Source-Level Debuggers by not mentioning what is perhaps the most important criterion for getting one: does the debugger allow access to all variables within a program, external static, internal static, local (or automatic as some people call them), as well as global. By access, I mean that all variables should be allowed to be viewed by a fairly simple naming scheme, not by some weird convention which requires the debugger (human) to figure out offsets from the stack. Except when I am interfacing assembler with C and must know how to

pick up variables being passed and to return a value, I do not want to worry about where off the stack things are, either in my program or debugging.

I know the Desmet debugger works admirably in this area, having used it on both CP/M-86 AND MS-DOS. No mention is made in the article at all about this area and issue. A C source level debugger that does not allow me easy access to all variables, especially local, is for me severely crippled if not almost completely worthless. That is why no one where I work (I'm a contract programmer) uses C-Sprite with the Lattice C compiler. I mentioned this deficiency to Lattice at Comdex in Las Vegas, but doubt they care too much. I care however and wouldn't waste my money on a source level debugger otherwise. I feel this should definitely have been mentioned in your article by Mr. Sachs.

Thanks,
Edward Diener

I agree completely with Mr. Diener's statement that a source level debugger should give access to all variables by name including locals and statics. I consider that to be part of the definition not only of a source level debugger, but of a symbolic debugger as well. All of the products reviewed could do so.

I did not mention this criterion because I considered it so basic and so obvious that there was no need. In the light of Mr. Diener's response, I regret that I was not more explicit.

Jonathan Sachs

GENERIC MS-DOS SOFTWARE

Dear MS/J:

It was not fair to dismiss Mr. Kasper's letter (Vol.2 No.2) concerning generic MS-DOS software saying that all you need is IBM-PC compatible video, keyboard, and disk hardware.

MS-DOS is evolving into a useful operating system with its support of device drivers and wide range of DOS functions. But, if you want to build a system where the standard input and output device is through a serial port, then you lose out on many applications.

Neither the software mass merchandiser, nor the software companies want to mess with such systems, so I think you could be of service if you could keep a list of truly MS-DOS generic products. In your Publishers' Report of the same issue, you say you want to cover topics of special interest to advanced users. I think this would fit in your charter.

Here is some software you can add to this list:

1) CompuView (v1.4, \$150) or Vedit+ (v2.32, \$225) and their Vspell and Vprint add-ons.

2) Micropro still has a generic version of Wordstar.

3) Cosmos has a version of their Pick-oriented Revelation database package.

4) Raima offers their db_Vista with source that is or can be made generic.

5) Lattice compilers offer very good separation of MS-DOS vs. PC-DOS functions in their library and their compiler is not PC-DOS specific.

6) Plink86 is an excellent linkage editor that runs MS-DOS. Their debugger, PfixPlus, fails.

Please reconsider your position and help in the support of these specialized system applications. Thank you for your time and keep up the good work.

Robert Allen
Garland, TX

DYNABYTE HELP NEEDED

We have a Dynabyte system and are in need of help; Dynabyte is *closed* and no service is available locally. Using the octaport board, we frequently experience having the data to any terminal (using 4), or to the printer, slow down to a snail's pace. When we reboot it's fine for a short time. When we Dynasys and take the octaport board out of the system (still on bus), everything works fine (2 term + parallel port). The octaport board has been replaced. Some say it may be the motherboard, any ideas?

Charles Baker
Associated Marketing Consultants
Echo Glen, P.O. Box 490
Gates Mills OH 44040

FROM A NEW CP/M + USER

Dear Bob,

You asked your readers what they are interested in. I recently started working with CP/M Plus and am interested in all the information I can get on it. I was a main-frame computer programmer for many years, but when I decided to buy one of my own, I went the inexpensive route and bought a Commodore 64. I recently upgraded to a Commodore 128 because it has a better CP/M package. I only found two books in the Wash D.C. bookstores on CP/M Plus. The documentation and BIOS source code Commodore sent me helped. CP/M on the C-128 only handles Commodore printers but not normal ASCII printers. I had to modify the printer driver to use the print capabilities of my Panasonic KX-1091. Commodore is supposedly releasing a new version of CP/M to satisfy customer complaints. I found out from Digital Research that the documentation for MAC, RMAC and LINK-80 is not part of the CP/M Plus documentation. I finally got a copy of it

BSW-Make

A practical and efficient

software configuration manager

for MS-DOS, VAX/VMS, and VM/CMS

At The Boston Software Works, we routinely work with a number of different operating systems and development environments. One tool we have found to be indispensable is BSW-Make. BSW-Make is a complete implementation of the UNIX *make* utility. It automates the tedious task of rebuilding your software product after an editing session; BSW-Make does only the minimum work required to update your product after a change, saving time and preventing missed compiles.

We carefully constructed BSW-Make to be portable, and have used it successfully under MS-DOS, PC-DOS, VAX/VMS, and VM/CMS. We wouldn't want to start a major software project without it, and we think you won't either, once you've tried it.

Highlights of BSW-Make:

- Works with any compiler, assembler, linker, or text processor
- Not copy protected
- Indirect command file generation facility overcomes operating system command length limitations
- Macro facility for parameterized builds
- Syntax compatible with UNIX *make*
- 30-day unconditional money-back guarantee

MS-DOS
\$89.95

VAX/VMS
\$495.00

VM/CMS
\$550.00/yr

BSW-Make for MS-DOS runs on any MS-DOS machine. It requires MS-DOS or PC-DOS version 2.00 or later, and is shipped on IBM PC 5 1/4 inch diskettes or IBM PC Convertible 3 1/2 inch diskettes.

BSW-Make for VAX/VMS runs on any VAX or MicroVAX running VMS version 4.0 or later. It is shipped on 9-track magtape or RX50 diskette.

(Available September, 1986)
BSW-Make for VM/CMS runs on any IBM 370-series, 43xx, 308x, or 309x system running VM/CMS. It is shipped on 9-track magtape.

All prices include shipping within the United States. Foreign orders add \$10.00 shipping and handling. We accept checks, MasterCard or VISA, or company purchase order.

The Boston Software Works, Inc.

120 Fulton Street, Boston, MA 02109
(617) 367-6846

when I bought a copy of PL/I-80. The local computer store copied the disks from 8" to 5 1/4" disk format for me. I'm now interested in modifying CP/M to optimize the directory and disk buffers for my configuration. The buffers in bank 0 do seem to speed up the disk access in some situations. I have one fast double sided drive (1571 410K) and one slow single sided drive (1541 170K). Putting all files on the fast drive is the fastest configuration. Using both drives with some files on each drive is the next fastest. Using the fast drive as both drive A and virtual drive E is the slowest configuration. In defining the keyboard characters, Commodore used some control key values to generate some ASCII characters not on the keyboard (i.e. ^ = is ~, ^: is {, and ^; is }). This is a problem for some people who want to use some of the rare control key values. Com-

modore's version of CP/M allows the keys to be redefined. Since ^9 and ^0 on the numeric keypad does not have any values assigned, I'm thinking of redefining them. CP/M Plus seems to have addressed many of the problems with CP/M 2.2. I like the quick warm boot and the ability of typing a series of commands on one line so batch files are not necessary in many cases. BIOS calls are a problem with CP/M Plus, but a lot of software that depend on BIOS calls is being modified to run under CP/M Plus. I'm trying to get copies of Dr. Dobbs Journal that had articles on CP/M Plus since not much literature is available on CP/M Plus.

Another of my interests is computer languages, and this is what got me involved with CP/M in the first place. I bought the MIX C compiler, the MIX
continued on page 26

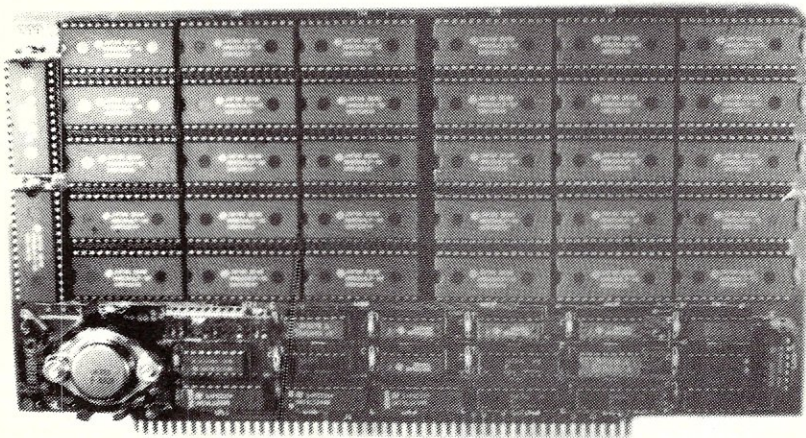
256K BYTES OF 100ns STATIC MEMORY

\$395

NO HIDDEN CHARGES!

Performics prices include dynamic burn in and UPS 2nd day air delivery.

* 128K byte version (depopulated 256K) \$345



S-100

STANDARD FEATURES

- 100ns memory chips
- Operates in excess of 12 MHz
- Certified system ready (industrial grade dynamic burn in)
- Supports 8 and 16 bit data transfers
- IEEE 696/S-100 compatible
- 24 bit addressing
- Address strappable to any 128K block within the 16 meg address range
- Extremely low power consumption
- Single +5 volt operation
- One year warranty

AVAILABLE SOON! High speed dynamic ram boards. 1 & 2 megabyte densities at Performics realistic prices.

PERFORMICS

INC.

Delivering high performance products and excellent service at realistic prices.

(603) 881-8334 ORDER NOW!

TERMS: Check, Visa/Mastercard

Performics guarantees our boards will work in your system or return for full refund.

11 Morning Dove Road • Kingston, NH 03848
• Hudson, NH 03051



A NEW ERA FOR HIGH PERFORMANCE TEST AND MEASUREMENT EQUIPMENT CONTROLLED BY PC'S

TURN YOUR PERSONAL COMPUTER INTO TEST AND MEASUREMENT EQUIPMENT. INSTANTLY!

There's been other hardware available, but if you really want to use it, you have to develop your own software. The others are just cards. You can't buy a piece of equipment, plug it in, slip in the disk and go to work. You're on your own.

Not with Rapid Systems. Rapid Systems Test and Measurement PC Peripherals are ready to go, the minute you receive them.

Plug the Peripheral into your personal computer, slip in the disk, touch one button and you're ready to work. It's that simple and easy.

Plus, all the post processing capabilities of your personal computer are at your fingertips. Store and retrieve waveforms from disk. Analyze and process the information. Your computer provides intelligent control and analysis, and it's available for complex mathematical computations and word processing.

FINALLY, INEXPENSIVE, INTELLIGENT AND HIGH PERFORMANCE TEST AND MEASUREMENT EQUIPMENT.

Take a look at the performance and capabilities of our line of test and measurement PC peripherals. You'll discover technology and high performance at prices you wouldn't have thought possible. A Digital Oscilloscope Peripheral for only \$599. An Enhanced Scope for \$725. Spectrum Analyzer for \$788. Data Logger or Data Acquisition Peripherals—your choice at \$688.*

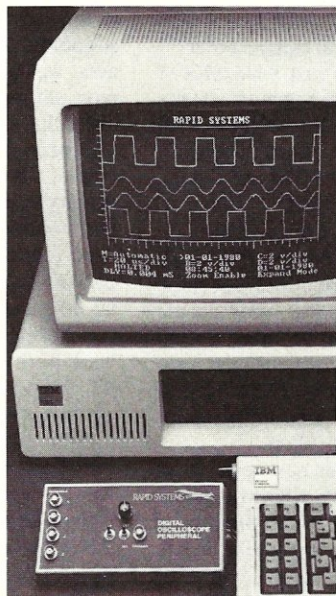
Powerful test and measurement equipment is as close as your personal computer for only a few hundred dollars.

DIGITAL OSCILLOSCOPE PERIPHERAL

For IBM, IBM compatible, Apple and Commodore personal computers. The peripheral hardware is contained in a compact 7.5"×4"×1.2" rugged ABS package that attaches to the computer via a ribbon cable and interface card. The digital oscilloscope software turns the personal computer into a versatile 4-channel digital oscilloscope.

Major features:

- Sampling rates to 2 MHz.
- Variable gain: 1.6v to 320v p-p.
- Resolution to 6.25 mV.
- 8-bit A/D conversion.
- Simple menu driven turn key operation from the keyboard.
- Multiple mode triggering.
- Enhanced graphics display.
- Diode protected inputs to 300 v.
- Storage/Retrieval of waveforms.

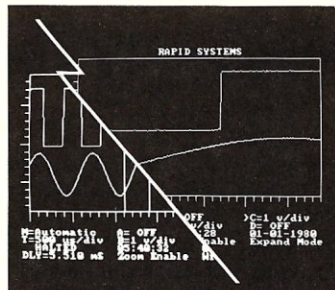


Rapid Systems Digital Oscilloscope Peripheral.

ENHANCED DIGITAL OSCILLOSCOPE PERIPHERAL

The enhanced digital oscilloscope is a software enhanced digital scope. The major features of this unit are:

- Zoom mode.
- Active scrolling of the waveform.
- Transfer data to 123/spreadsheets, dBase files, or spectrum analyzer.
- Assembly language speed.
- Help file.
- Printer drivers.

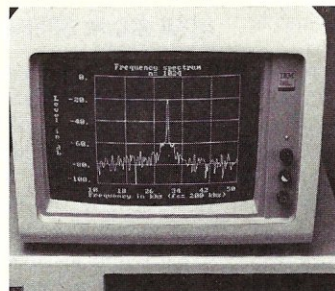


Zoom Mode of Enhanced Digital Oscilloscope Peripheral.

SPECTRUM ANALYZER PERIPHERAL

The Spectrum Analyzer Peripheral converts your PC into an instrument that characterizes an electrical signal for its frequency content. The peripheral uses the Fast Fourier Transform (FFT) in software to compute the signal spectrum. Major features:

- 2 MHz digital scope software included free.
- Sampling rates to 500 kHz.
- Menu driven operation.
- Store/Retrieve/Print modes.
- Amplitude versus Frequency display using enhanced graphics.
- Hardware and Software triggering.
- FFT sizes: 16 to 1024 points.
- Frequency/amplitude cursor.
- Spectrum averaging.



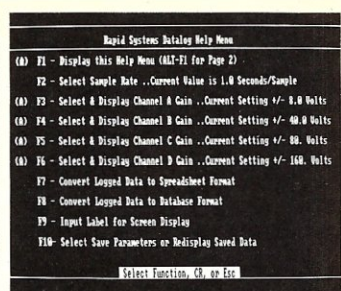
Rapid Systems Spectrum Analyzer Peripheral.

DATA LOGGER PERIPHERAL

The Data Logger Peripheral allows the IBM or compatible personal computers to be used as versatile 4-channel data logging systems. The peripheral provides continuous measurement of up to four channels of data at a user selectable rate, and the data may be put out to hard disk or floppy. The number of data points to be taken is selectable by time or number of samples. Additionally, Start may be driven by either time and/or event (an internal or external trigger).

For your copy of the new Rapid Systems Test and Measurement Peripherals catalog, to order, or for further information, call or write Rapid Systems, 755 N. Northlake Way, Seattle, WA 98103. (206) 547-8311. Telex: 265017UR.

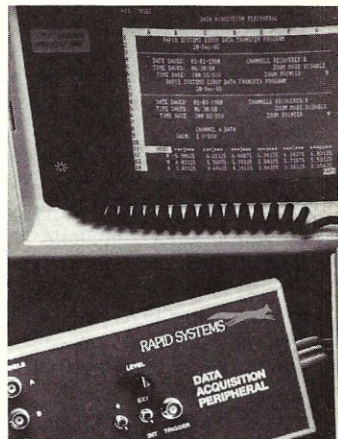
RAPID SYSTEMS



Help menu from Data Logger Peripheral display.

Major features:

- 8-bit A/D conversion.
- Diode protection on all inputs.
- Sampling rates from .1 second to 999.9 seconds or event driven.
- Simple menu driven operation from the PC keyboard.
- Graphics display using 160×488 pixels for data display and data windows for parameter selection.
- Storage/retrieval of data on disk.
- Strip Chart printing with graphics printer.
- User selectable display headings.
- Automatic scrolling of a displayed waveform.
- Data reduction utility source and object code included!



Display from Data Acquisition Peripheral.

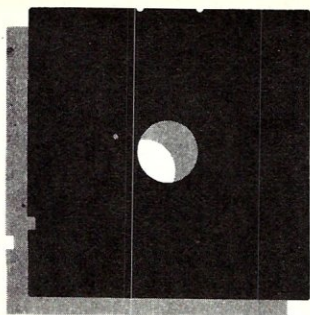
DATA ACQUISITION PERIPHERAL

The data acquisition peripheral allows the user to customize the PC for a specific data acquisition application. This unit provides a high speed, easy-to-use software interface for direct data access and control of the hardware. The User Definable Interface software (UDI) included with the peripheral provides assembly language subroutines dedicated to the control and acquisition of data from the hardware.

Major features:

- Digital oscilloscope software included free.
- Sampling rates to 2 MHz.
- Data acquisition setup utility.
- Data transfer utility.
- Selectable gain/sampling rate.
- Trigger mode selection.

*Prices are for IBM compatible peripherals. Other models cost less. Additional software and accessories available.



In the SIG/M Public Domain

by Stephen M. Leon



If I could make three wishes of programs I would like to see contributed to the public domain, what would I pick? That was the question Hank Kee posed the other day. I have been pondering the answer since then.

Let's face reality, you probably cannot do, in the present CP/M-80 environment, what you can do in either CP/M-86 or PC-DOS. But do you really have to do most of those things? The answer, sadly, is yes!

My CompuPro system has 512K. One 256K Ram-22 board for the machine cost me as much as a 640K loaded PC clone now costs. But, when I run Concurrent on the CompuPro I effectively have four separate computers and can switch back and forth between them. Sure, when I run CP/M 80 on the CompuPro, all that extra memory becomes an M-Drive ram disk...but so what. For less than \$400 I added a 2 meg JRAM-3 to the PC, and now have one heck of a ram disk. What is to become of all those CP/M-80 machines is the area of my wishes.

What heads my wish list? CP/M+ never caught on because Rich Conn's ZCPR brought to CP/M just about everything that Plus added. Well just about everything. Heading the wish list is ZCPR4. The man who has kept CP/M alive long after it should have been six feet under is about the only person I can think of who might come up with an easy way to increase the memory of CP/M-80 and, at the same time, add some type of concurrency. Included would be something akin to the Intel EMS specification that would allow the ZCPR utilities and other programs to be parked and swapped to a ram drive...kind of like the way DESQview does it on the PC.

No way this can come as a commercial product either. If it is to be made available to all, it's going to have to be the last gasp of the CP/M hacker fraternity. It isn't really that hard to do...I know someone who has already modified his CP/M 2.2 BIOS to do most of this, so how about it Rich?

The second item on the list also relates to Rich Conn. One of the things that has expedited the expansion of the PC-DOS world is the availability of free or cheap

applications software. Rich is doing a great deal of work on the Department of Defense ADA library. I understand from him that the DOD ADA library contains all sorts of applications software. All of it is available for free with a *big but*. The *but* is that there is really nothing available to compile ADA on a CP/M machine. There is an ADA subset, Little Ada, on Volume 92, which could serve as a starting point for a full compiler. Wish two then is for someone to write and donate a full ADA compiler to the SIG/M library.

The last item on the wish list is also well within the realm of possibility. There is software now that can do some of it, but I would like to see someone put it together as a complete public domain package. The program: a way to run a CP/M machine off a serial port of a PC as a second console for the PC with full emulation of the PC screen and keyboard and without a significant speed loss on either machine.

HOW SIG/M OPERATES

Time to stop dreaming and go back to the world of reality. Micro/Systems Journal generates a lot of SIG/M mail. We especially appreciate the software donations that are coming in. Please keep them up. We also have some complaints, with the most common one being "why does it take so much time for SIG/M to ship an order?"

First, let me explain just what SIG/M is. Legally, it is a user group of the Amateur Computer Group of New Jersey. As a practical matter, it is a not-for-profit cooperative effort by people throughout the world. It is strictly a volunteer organization. Dave Wrobel is the SIG/M chairman. Dave Raibert is the Treasurer. As Disk Editor, I review the contributions that come in, reject some, accept some, prod people to write programs where we think a need exists, and then organize them into volumes of approximately 230K each. I have the fun part of the job, sort of a kid with free run in the candy shop.

We normally date our new releases as of the date of our monthly SIG/M meeting (usually the third Friday of the month). If all has gone well, some two weeks before the meeting, we prepare master copies of

Steve Leon is the SIG/M Disk Editor. In other words, he is the person who assembles, compiles, and edits all of the SIG/M public domain software disks. Thus, he speaks with the greatest authority as to what is going on in the SIG/M public domain software area.

the new volumes to Bob Todd, who is in charge of SIG/M distribution. At the same time we air mail copies to selected foreign distribution points (England, Japan, Australia and Mexico). Bob makes additional copies of the disks and starts them off on a series of circuits that reach all of the US and Canadian distribution points. We also prepare the seed copies for mail order distribution, and make copies for the ACGNJ and NYACC BBS systems and CompuServe. Thus, if all goes well, the software is available throughout the worldwide network by the release date.

SIG/M mail order distribution is handled by Glen Dusch. Glen picks up the mail at the SIG/M box (Box 97, Iselin NJ 08830) and sees to it that mail orders are filled as soon as possible. It normally takes about a month from the time he receives an order at Box 97 to the time the UPS man delivers it to the user. Up to now, SIG/M itself only supported 8" SSSD format disks. However, we have started an experiment and are now shipping all SIG/M volumes on 5" disks that can be written on the PC using Uniform, (just about all formats except Apple, Commodore 64, or any of the high density formats). These disks are \$7.00 per volume. However, for formats that have less than 240K per volume (usually SSSD formats), there will be an additional \$2.00 charge.

We encourage users to get their software from places other than SIG/M at Box 97. We do our best to get the new volumes out to the network so that you can get them locally. Many club user groups also make them available for a nominal copying charge (usually \$1/volume) with the user providing the formatted disk.

Delays in shipping SIG/M orders are usually caused by such factors as waiting

for catalogs to be printed, everyone being tied up with the Trenton Computer Festival, Glen going on vacation, etc. And, when you add complications to a SIG/M order, such as requesting information or asking us to pick volumes for you, or asking for one program from each of 17 disks or not being sure of a volume number, all too often the order winds up on a pile waiting for someone to stop over at Glen's place to answer some mail. A few years ago the SIG/M Treasurer blew a fuse when he found a stack of stale checks for filled orders, so now Glen usually processes the checks early in the game.

The bottom line is: if your order isn't filled within a month, something is wrong and we want to know. Drop a note to SIG/M, Box 97, Iselin NJ 08830. Tell Glen what you ordered and when. If you want to complain to me and have me call up Glen, drop me a note at 200 Winston Drive, Cliffside Park, NJ 07010. Let us know if there is a problem. But don't be too harsh on us, because we do try, but even in an organization of volunteers, there is no excuse for doing a poor job.

NEW SIG/M RELEASES

In the last issue we introduced Volume 255 as the first of a number of new releases from the Japan User Group. We now have seven additional volumes and more are expected shortly.

Volumes 265 to 268 constitute a four volume General Activity Simulation Program (GASP). Written in Fortran with code for both CP/M and CP/M 86, this series is duck soup for the queuing crowd. Volume 268, in Fortran 80, is Statistical Distribution and Multiple Regression.

These programs are obviously geared to a specific area of user, but they do hold some general interest for all who are interested in the type of coding that is coming out of Japan. Some of the routines are quite interesting and readily adaptable by Fortran programmers. Aside from that, as someone who wound up writing a queuing routine for the student cafeteria as a term paper, you never know how you can adapt from the public domain to meet your specific needs.

There are a number of other statistical programs in the SIG/M library. These include a statistical analysis package in Pascal-Z on Volume 71 and statistics in North Star Basic on Volume 182. There is another simulation model on Volume 123.

I took another look at the program on Volume 123. It is a 47K file called ODMP.LBR. According to the program documentation it provides an operational environment similar to an Analog Computer using a block diagram approach and a second order Runge Kutta (modified Euler) integration algorithm. If you know what that means, you may be interested in the program.

What did come up in this review, and of general interest to all, is the difference between library utilities. Over the years LU (library utility) has been much improved. The NULU version does even more than just plain LU, but one important change in version 3 of LU is the addition of a CRC checklist in the file. If you have a library created by the old LU and you try to read it with the new LU, you will get a notation of a CRC error. Use the -L command to list the contents and you will see that all CRCs are 0. It does not mean that there is an error in the file. It means only that the library was created with the old LU. (I should also note that the LU86 we put on CP/M 86 disks with library files also does not support CRC.) If you want to add the CRC values to an old library, merely open up the file with LU301 (command -O). Then reorganize (command -R) with the same number of entries. Before you do this, however, I suggest you verify the CRC of the entire library with the CRC in the catalog file for the volume. That way you know that the file has started out with the proper CRC.

On the subject of CRC, sometimes a change in format produces a change in CRC value. The catalog files on SIG/M volumes are automatically created by John Rath's CRCBUILD. The CRC values for each program are stored in the catalog. Glen tells me that sometimes when he transfers a disk from standard 8" to one of the 5" formats the CRC value of the file changes because of the way the format handles such things as blank spaces. I have tried to duplicate this, but so far with no luck. Therefore, if you have a bad CRC on a 5" disk, you may still have a good program.

Someone called the other day looking for a program to unprotect an MBASIC file. (The easiest way to find something in the library is to use the catalog volume, Volume 0 and do a string search.) Anyway, if you have the same problem, unprotect for MBASIC is on Volume 52. MBASIC users may also be interested in VMAP on Volume 40, an MBASIC variable mapper, and MXREF on Volume 78, an MBASIC cross reference utility.

SIG/M Volumes are available on 8" SSDD Disks for \$6 each (\$9 foreign) from SIG/M, Box 97, Iselin, NJ 08830. Volumes may also be ordered in most 5" formats (other than Apple, Commodore 64 or high density format) for \$7 each (\$10 foreign). There is an additional charge of \$2 for formats which require more than one disk to hold a 240K volume (such as SSDD formats). Printed catalogs are \$3 each (\$4 foreign). Disks in a variety of formats may also be obtained through the worldwide SIG/M distribution network. The distributor list is included with the printed catalog. A disk version of the catalog (Volume 00) is available for \$6. It also contains the distribution list. Many bulletin boards have the software for downloading and most new releases are available on the CP/M Sig on Compuserve. (M)

tin boards have the software for downloading and most new releases are available on the CP/M Sig on Compuserve. (M)

NEW PUBLIC DOMAIN RELEASES

Volume 265

GASP (General Activity Simulation Program) from Japan User Group Volume 7 (Vol 1 of 4). Documentation and CP-M 80 Version

Volume 266

GASP (General Activity Simulation Program) from Japan User Group Volume 8 (Vol 2 of 4). CP-M 86 (SSS-Fortran or ai-Fortran) Version and Source Programs of GASP Subroutines

Volume 267

GASP (General Activity Simulation Program) from Japan User Group Volume 9 (Vol 3 of 4). Executable GASP Code for CP-M 86

Volume 268

GASP (General Activity Simulation Program) from Japan User Group Volume 10 (Vol 4 of 4). Extended Version of GASP II for F80 & F86

Volume 269

Statistical Distribution and Multiple Regression with Interval Estimation for Fortran-80 from Japan User Group Volume 12

Volume 270

Turbo Pascal Programming From Japan & Abstracts from various Japan User Group Volumes

Volume 271

C Programs from Japan & Abstracts from Japan User Group Vols 19 & 20

FREE

52 PAGE CATALOG AND APPLICATIONS GUIDE

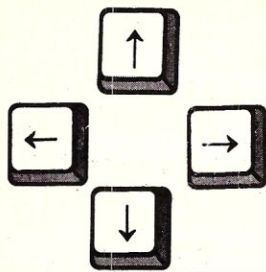
AFFORDABLE ENGINEERING SOFTWARE

CP/M MS-DOS TRSDOS PC-DOS

<p>PDP — Plotter Driver Program For multi and single pen plotters \$72.95</p> <p>XFER — Transfer Function Synthesis Transfer functions / Electronic circuits \$72.95</p> <p>TEKCALC — Scientific Calculator Program Screen graphics / Statistics / Programmable \$72.95</p> <p>COMCALC — Communications Design Spreadsheet Communications budget calculator \$72.95</p> <p>RIGHTWRITER — Report Proofreader Program Version 2 — Applies 2,200 rules of English \$97.95</p>	<p>ACNAP — AC Network Analysis \$72.95</p> <p>DCNAP — DC Network Analysis \$72.95</p> <p>SPP — Signal Processing Program \$72.95</p> <p>PLOTPRO — Scientific Graph Printing \$72.95</p> <p>PCPLOT — High Resolution Graphics \$72.95</p> <p>LOCIPRO — Root Locus Analysis \$72.95</p> <p>ACTFIL — Active Filter Design \$72.95</p> <p>STAP — Static Thermal Analysis \$72.95</p> <p>MATRIC MAGIC — Matrix Manipulation \$72.95</p>
--	--

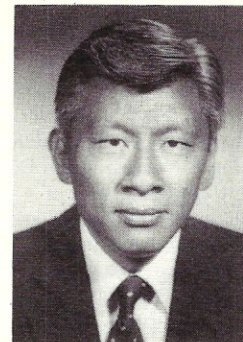
BV Engineering
Professional Software
2200 Business Way, Suite 207, Riverside, CA 92501 U.S.A.

(714) 781-0252



PC/Blue Report

by Hank Kee



There are now 200 volumes in the PC/Blue Public Domain Software Library. All due to the efforts of many unsung heroes who contributed their time and talent to develop these programs and make them available as either public domain or user-supported software. The library continues to grow dramatically.

The PC/Blue library represents a very rich source of software for the PC and compatibles. This software collection, over the past four years, has enabled the new user to build a basic software library at minimal cost. Many computer clubs also use the library to attract members as well and cover operating expenses by charging a nominal amount for copying (typically \$1 to \$2 per volume) for copying.

I have often been asked, how good are the programs? They are, for the most part, very good. Some of the programs have no commercial equivalent and cannot be found anywhere else. Public domain and user-supported programs were made available through this medium of distribution because normal outlets for aspiring software entrepreneurs no longer exist.

How would one go about selecting from this vast repertoire of software? I will try to point to, what I feel are, the best of the collectibles, in terms of a "hacker's pack", a "businessman's pack", and a "fun pack". I have assembled each pack into a ten volume set. The number of volumes in a set is arbitrary. A box of diskettes normally contain ten.

Fundamental to all the packs, is a good modem communication package. Hackers, game enthusiasts, and business users alike need to communicate with other computer systems. *QModem* is clearly the most popular of them all. It is easy to use, functionally comprehensive, well maintained and richly documented. An excellent rival to *QModem* is *ProComm*. *ProComm* is very similar to *QModem* in basic function but has many more features.

THE HACKER'S PACK

The components for the Hacker's Pack, should include good program development and artificial intelligence languages, an expert based system shell, and tools for testing PC's. And, of course, what would a hacker be without CP/M?

Recommended Hacker's Pack

QModem (vol 191)/ProComm (vol 200)	modem communications
RBBS-PC (vol 198-199)	remote bulletin board system
LISP (vol 188)	Lisp interpreter
PD Prolog (vol 183)	language processor
Small C (vol 154)	original "small" C
CP/M 2.2 Emulator (vol 185)	CP/M 2.2 emulator for V20 chip
The Expert (vol 176)	expert system shell
PC-Write (vol 167)	word processor
PC Magazine Lab Series#1 (vol 135)	benchmark programs
PC Magazine Lab Series#2 (vol 179)	benchmark programs

THE BUSINESS PACK

The businessman needs his standard fare of word processing, spreadsheet and database management systems. Add to this some of the desktop management tools along with an investment record system and you have a pretty comprehensive generic business pack.

Editor's note: Hank Kee is the librarian for the PC/Blue public domain software library. He is the person who collects, assembles, and checks all the software issued by PC/Blue and then compiles and edits them into the released volumes.

Recommended Business Pack

QModem (vol 191)/ProComm (vol 200)	modem communications
PC-Calc (vol 141)	electronic spreadsheet
PC-Write (vol 167)	word processor
Personal Management (vol 133)	DeskMate/Partner
pBase (vol 168)	database management system
PC-DBMS (vol 131)	database management system
DOS Assistant (vol 162)	operating system support
TAM (vol 113)	financial monitoring system
Stock (vol 120)	investment record system
DBS-Kat (vol 197)	super diskette catalog system

THE FUN PACK

The fun pack is for those looking for something unusual. This should include graphics and some smart and slick graphic games to make up this spiffy set.

Recommended Fun Pack

QModem (vol 191)/ProComm (vol 200)	modem communications
WPK (vol 95)	word processor for kids
Present (vol 164)	slide presentation system
EScreens (vol 171)	create interactive screens
Origami (vol 132)	Japanese art of paper folding
3By5 (vol 125)	information management
GluDraw (vol 85)	line graphics system
DBS-Kat (vol 197)	super diskette catalog system
CaveQuest (vol 178)	dungeon/dragon graphics game
Striker (vol 190)	chopper graphics game

The recommended lists are, at best, personal preferences. There are many other well deserving software systems which could have easily added to the *recommended* lists. Software is organic. By that I mean that there will be continual changes. (μ)

NEW PC-BLUE RELEASES

The following are the new releases to the PC/Blue library:

Volume 188

LISP Interpreter by Marc Adler
NYWord version 1.2-word processor by Marc Adler

Volume 189

Color Graphics Adapter Games
One-Arm BANDIT, BlackJack-BJ, Poker, Rockets, Roulette, Stock & SubChase

Volume 190

Color Graphics Adapter Games
Striker, Kong, Golf & Monopoly

Volume 191

QModem version 2.0E - modem communications by John Friel III

Volume 192

ICON-high level language version 5
SNOBOL like programming language by Ralph E. Griswold
Small-C Interpreter version 1.3 by Bob Brodt

Volume 193

Colossus-remote bulletin board by Forbin Project/Dan Plunkett
QModemJr-modem communications by Forbin Project/John Friel III

Volume 194

Kermit version 2.28 - modem communications

Volume 195

EZ-FORMS - form generating program
PC-Font2 version 2.04
Epson printer utility

Volume 196

Nutrient-food analysis program

Volume 197

DBS-Kat version 1.3 - super capacity diskette cataloger by Applied Foresight Inc.

Volume 198-199

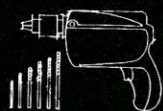
RBBS-PC version 14.1A - remote bulletin board system

Volume 200

ProComm version 2.3 - modem communications by Bruce Barkelew/
Tom Smith

Copies of the PC/Blue printed software directory can be ordered from Micro/Systems Journal, Box 1192, Mountainside NJ 07092 (\$5 U.S., Canada & Mexico; \$7 foreign).

PC/Blue disks are available from the New York Amateur Computer Club, Inc., Box 106, Church Street Station, NY NY 10008. Price is \$7 per volume which includes media, postage, and handling. On Foreign orders, please add \$2 per disk.



Power Tools for system builders™

Call today for our free catalog of design aids, compilers, libraries, debuggers, and support tools for Apple and IBM micro computers. The Power Tools catalog includes product descriptions, warranty and license terms, and all the information you need to make an intelligent purchase decision.

TSF offers technical support, competitive pricing, free UPS shipping on orders over \$100, and a reasonable return policy. Visa, MasterCard, and American Express accepted without surcharge. **TSF helps you get your job done.**

Sample Prices:

Microsoft C \$259
MASM 4.0 \$109
Turbo Pascal \$45
Mark Williams C \$375
Lets C \$59
Wendin OS Toolbox \$89
Blaise Async Manager \$137

Call Toll Free

24 hrs a day/7 days a week

Ask For Operator 2053

800-543-6277

Calif: 800-368-7600

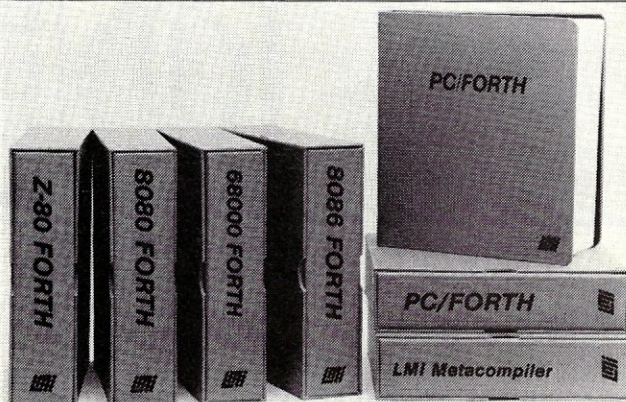


TSF

The Software Family™

• Dept. C-2 • 649 Mission Street
• San Francisco • CA 94105
• (415) 957-0111

TOTAL CONTROL with LMI FORTH™



For Programming Professionals: an expanding family of compatible, high-performance, Forth-83 Standard compilers for microcomputers

For Development:

Interactive Forth-83 Interpreter/Compilers

- 16-bit and 32-bit implementations
- Full screen editor and assembler
- Uses standard operating system files
- 400 page manual written in plain English
- Options include software floating point, arithmetic coprocessor support, symbolic debugger, native code compilers, and graphics support

For Applications: Forth-83 Metacompiler

- Unique table-driven multi-pass Forth compiler
- Compiles compact ROMable or disk-based applications
- Excellent error handling
- Produces headerless code, compiles from intermediate states, and performs conditional compilation
- Cross-compiles to 8080, Z-80, 8086, 68000, 6502, 8051, 8096, 1802, and 6303
- No license fee or royalty for compiled applications

For Speed: CForth Application Compiler

- Translates "high-level" Forth into in-line, optimized machine code
- Can generate ROMable code

Support Services for registered users:

- Technical Assistance Hotline
- Periodic newsletters and low-cost updates
- Bulletin Board System

**Call or write for detailed product information
and prices. Consulting and Educational Services
available by special arrangement.**



Laboratory Microsystems Incorporated

Post Office Box 10430, Marina del Rey, CA 90295

Phone credit card orders to: (213) 306-7412

Overseas Distributors.

Germany: Forth-Systeme Angelika Flesch, Titisee-Neustadt, 7651-1665

UK: System Science Ltd., London, 01-248 0962

France: Micro-Sigma S.A.R.L., Paris, (1) 42.65.95.16

Japan: Southern Pacific Ltd., Yokohama, 045-314-9514

Australia: Wave-onic Associates, Wilson, W.A., (09) 451-2946

YOUR EAST COAST HI-TECH CENTER

**QUALITY
COMPUTER
PRODUCTS**



**DIGITAL
DECISIONS
SYSTEMS**

**FOR THE
BUSINESS
PROFESSIONAL**

Compu Pro/Viasyn-\$100

CPU 286-8 3CY	542	SYS.SPT.1	319	RAM 23 128K	262
CPU 286-8 2CY	674	INTF 4	333	RAM 24 1MEG	2249
CPU 86 -10	295	DISK 1A	445	MDRIVE/H.5M	366
CPUZ-8	199	RAM 22 256K	446	MDRIVE/H.2M	819
SPUZ-256	289	RAM 23 64K	187	INTF 3-8	445
				DISK 3	520

LOMAS MEGARAM 2-4 MEGABYTES

MODIFIED FOR USE WITH DISK 3 SYSTEMS

ADM 12+ 2 PAGES

TY1925 COMP

LINK 125 2 PAGES

WY50 STYLE

489

MAXTOR

1085

1889

TANDON TM755

945

We can custom configure any system to meet your needs. Installation & training available in some areas.

WATCH FOR OUR 60 MEG S100 TAPE BACKUP...

CALL!!

P.C. UPGRADES

MF-384 384K RAM & SER/PAR/CLK/GAME	229
MULTI 1/0 SER/PAK/CLOCK HALF CARD	169
QUAD 1/0 4 RS232 SER FOR PC OR AT	199
SYNCO TTL MONITOR - TILT/SWIVEL - A OR G	135
TOSHIBA 360K FLOPPY W/MTG HDWE	95

**HARD
DISKS**

TULIN

26 MEG PC

389

40 MEG PC

499

SEAGATE 10 MEG PC

CALL FOR PRICE

CONTROLLER FOR ABOVE - W/CABLES

179

2-7 RLL HARD DISK CONTROLLER

289

INCREASE SPEED AND CAPACITY BY 50%

ADAPTEC 2070

LIMITED IBM PC W/

512K RAM • MONO TTL

\$1995

2 • 360K FLOPPYS SER • PAR •

CLOCK • GAME • VIDEO • DOS

\$2250

DD8000 series Professional Computer

**8 MHz V-20 CPU Provides Superior Performance
At A Reasonable Cost!**

STANDARD FEATURES INCLUDE:

- 1 MEGABYTE RAM
- 2 SERIAL PORTS
- 6 EXPANSION SLOTS
- MONO TTL MONITOR
- 2 PRINTER PORTS
- OPTIONAL HARD DISKS
- IBM's PC-DOS 3.1
- -20 TO 56 MEG
- 2 360K FLOPPY'S
- SASI PORT
- ADVANCED COLOR/MONO/HERCULES
- GRAPHICS DISPLAY ADAPTER

22 MEG \$2639

33 MEG \$2889

56 MEG \$3779

\$1899

WILL SUPPORT UP TO THREE PC-DOS USERS WITH CCP/M-XM

FLOPPY
ONLY

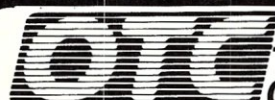


**Multitech
POPULAR 500
COLOR PC SYSTEM
\$1299**

- 2 360 K FLOPPYS
- 512K RAM
- PARALLEL PORT
- DOS 2.11

- RGB MONITOR
- RS232 SERIAL
- GAME PORT
- WORD PROCESSOR

QUALITY AND ECONOMY IN 16 BIT COMPUTING



**OUTPUT
TECHNOLOGY
CORPORATION 700
CPH!**

High Speed Matrix Printers

MODEL OT-700 \$1695

OT 700e \$1895

BULK 5.25" DISKETTES By BROWN

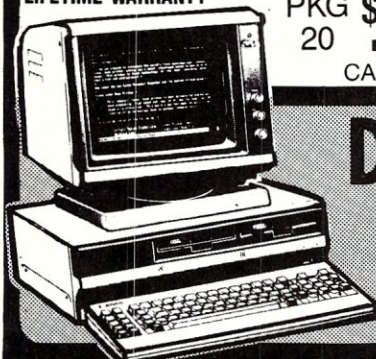
CERT. DS/DD W/ HUB RINGS & TYVEC SLEEVES

LIFETIME WARRANTY

PKG \$**.90**ea. BOX \$**.80**ea.

20 .90ea. 160 .80ea.

CALL FOR VOLUME PRICING



GIFFORD MC-DOS USERS MC-DOS ENHANCEMENTS!

UPGRADE YOUR MC-DOS SYSTEM
TO SUPPORT HIGH CAPACITY HARD DISKS

75 TO 330 MEGABYTES

PATCH KIT 1 — SOFTWARE ONLY \$100
(YOU SUPPLY HARD DRIVE)

COMPLETE KITS AVAILABLE — \$ CALL \$

STANDARD KITS ARE CONFIGURED FOR

MAXTOR 1000 SERIES DRIVES.

CUSTOM CONFIGURATIONS ARE AVAILABLE.

REQUIRES DISK 3 CONTROLLER SYSTEM

DIGITAL DECISIONS SYSTEMS

229 LAKEVIEW AVE • CLIFTON, NJ 07011

1-800-862-6233

Wait for tone,
dial Ext. 0014762

VISA

MasterCard

Discover

ALL PRICES SUBJECT TO CHANGE — SHIPPING EXTRA

IBM is a Reg. TM of International Business Machines. CP/M, Concurrent DOS are Reg. TM of Digital Research Inc. MSDOS is a TM of Microsoft. CompuPro is a Reg. TM of Viasyn Corp. OT 700 is a Reg. TM of Output Technology Corp. Popular 500 is a Reg. TM of Multitech Int.

THE TOOLS YOU'VE ALWAYS NEEDED But never have the time to write

FIND IT *FAST* WITH SEARCH! \$30

Available for both MS-DOS and CP/M, SEARCH can find a letter you wrote on Dec 24th about Widgets. SEARCH can find all the files in which you used 1985 that you'll need to change to 1986. SEARCH can find all the files in which you used a variable or called a particular subroutine. **FAST!!** And only \$30.

SEARCH for more than one string at a time. SEARCH multiple files & disks, use wildcards. See a window around the lines that contain the matches. SEARCH WordStar paragraphs, ignoring soft-carriage returns and soft hyphens.

SEARCH has on-line help whenever you need it. Send output to the screen, a file, or directly to your printer. The MS-DOS version supports path names.

SEARCH is NOT copy-protected. SEARCH is only \$30!

For a limited time, get both CP/M & MS-DOS versions for only \$55.

BIG MACHINE UTILITIES FOR CP/M \$45

CompuMagic's **Utility Package** will make your life easier, regardless of what you use your computer for. These programs simply do more and work better than the programs you're currently using to do the same things. Written by the wizard of CompuMagic, who insists on the same ease of use on his CP/M machines that he has on the big machines he uses during the day.

For only \$45, you get 20 super programs:

Seven file management (copy, rename, compare, etc) tools with a common command interface: command line and prompt modes; what-if, query, & overwrite modes; multiple commands on a line; rename of new files with wildcards; input redirection; changing source and destination user areas; concatenation; I/O to files and devices; etc.

Directory programs that take multiple file specifications; sort by name or file type.

Create short programs to substitute for long command lines.

Capture everything going to the screen (CP/M2.2 only), even through BIOS, and write a copy to a file.

MUCH MORE! ONLY \$45!

30 DAY MONEY-BACK GUARANTEE. Both SEARCH and the **Utility Package** come with CompuMagic's full, 30-day, money-back guarantee.

ORDER NOW: Send Check or MO (Personal checks accepted) or call with your MC, Visa, or COD order. NO charge for UPS ground shipping. Add \$2 for UPS COD and \$5 for overseas. (MD Res. +5% tax). Be sure to specify operating system and disk format.

Please Note
New Address
and Phone!!

CompuMagic, Inc.[™]
P.O. Box 437
Severn, MD 21144
(301) 969-8068

What you see is what you get . . . and send!

Transfer Protocol: Modem7/CRC Packet Size: 128 Files: 1

Block #	of	Kbytes	%	Time Remaining	Consec	Errors File	Total	Total Kbytes
31	522	3	5	5:06	0	0	0	3

Errors:

Status: Transfer in progress

[MEX File Transfer]



[CTL-C to abort]

Sending: >ANYFILE.AQC

Announcing Version 1.6 of **MEX**, the communications software with a view from the top. Regardless of your level of sophistication, **MEX** can put you on top of the data transfer game and keep you there. For the executive on the go, our new pull-down transfer screen and easy-to-use menus reduce the complexities of modem communications to a few keystrokes. For the advanced user, **MEX**'s greatly enhanced script processor offers a complete programming language for development of highly secure custom applications. If communication is money in your business, **MEX** may be the best investment you make this year.

Two options available:

MEX-PC is the most complete modem software you can buy. Allows you to switch between menu-driven and command-driven communications at will. Makes full use of Hayes AT command set, with overlays available for most other modems. Features include: complete script processor programming language; user-definable keystrokes; auto-dial and auto-baud-set phone libraries; all popular protocols, including MODEM-7 batch transfers. **\$59.95***

MEX-PAC — All the features of **MEX-PC**, plus:

- (1) A remote module that allows you to run your office computer from home, and vice versa; and
- (2) Terminal emulation that lets your PC masquerade as a DEC VT52/100 or Televideo 925 terminal for on-line communication with mainframes. **\$99.95***

XMODEM CRC/XMODEM CHECKSUM/KERMIT/COMPUERVE A/128 or 1K BLOCKS

Versions available for IBM-PC and compatibles, Tandy 2000 and most CP/M machines.

MasterCard, VISA welcome

**Give us a toll-free call at
1-800-NITEOWL**

In Wisconsin, 1-414-563-4013

* plus shipping: Wisconsin residents add 5% sales tax.

NightOwl Software, Rt.1 Box 7, Ft. Atkinson, WI 53538

DATA ANALYSIS AND GRAPHICS FOR SCIENTISTS AND ENGINEERS

SIMPLE. POWERFUL. FAST.

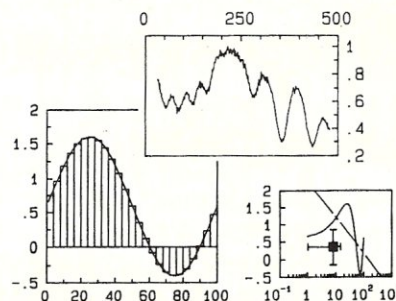
PLOTZ is a comprehensive data analysis and presentation graphics program designed specifically for technical applications. **PLOTZ** works with any ASCII data files, as well as spreadsheets. The program is entirely menu-driven, and features a command-file mode for automatic integration with other applications. **PLOTZ** allows unlimited overlays of plots and datasets, multiple ordinate and abscissa scales, multiple plots/page, lin/log scales, over 100 plot symbols, 7 linestyles, error bars, histograms, linear and polynomial fits, etc.

DATA OPERATIONS

F1 GET/STORE DATA	F2 EDIT DATA
F3 LINEAR SCALING	F4 LINEAR FIT
F5 SMOOTH	F6 POLYNOMIAL FIT
F7 DERIVATIVE (dy/dx)	F8 HISTOGRAM FIT
F9 PLOT MENU	F10 EXIT

PLOT OPERATIONS

F1 SPECIFY AXES	F2 ADD TEXT
F3 DRAW AXES	F4 ADD SYMBOLS
F5 DRAW DATA	F6 ADD ARROWS
F7 SCREEN PLOT	F8 SYSTEM SETUP
F9 DATA MENU	F10 EXIT



PLOTZ GRAPHICS runs on IBM PC/XT/AT and compatible microcomputers with 320K RAM and a serial port, and produces graphics output on the screen and on HPGL plotters.

Price: \$350/copy (\$300, academic). A full-featured demonstration disk and users' manual are available at no charge to qualified professionals.

CURTIS TECHNICAL SOFTWARE CORPORATION
PO Box 178 Pennington, NJ 08534 (609) 771-0875

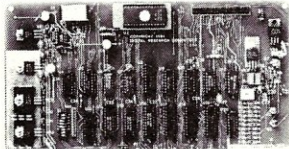
PLOTZ GRAPHICS

DIGITAL RESEARCH COMPUTERS

(214) 225-2309

\$100 EPROM PROGRAMMER

OUR NEWEST DESIGN, FOR FAST EFFICIENT PROGRAMMING OF THE MOST POPULAR EPROM'S ON YOUR \$100 MACHINE. COMES WITH MENU DRIVEN SOFTWARE THAT RUNS UNDER CP/M 2.2 (8 INCH). PC BOARD SET CONSISTS OF (\$100) MAIN LOGIC BOARD REMOTE PROGRAMMING CARD AND SIX PERSONALITY MINI BOARDS FOR 2716, 2532, 2732, 2732A, 2764, AND 27128. SOLD AS BARE PC BOARD SET ONLY WITH FULL DOC. SOFTWARE FEATURES "FAST" PROGRAMMING ALGORITHM. FOR Z80 BASED SYSTEMS.



PC BOARD SET, FULL DOCUMENTATION, 8 IN. DISKETTE WITH SOFTWARE.

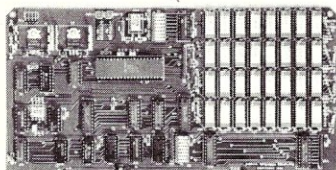
NEW! \$69⁹⁵

128K \$100 STATIC RAM/EPROM BOARD
JUST OUT! USES POPULAR 8K X 8 STATIC RAMS (6264) OR 2764 EPROMS. FOR 8 OR 16 BIT DATA TRANSFERS! IEEE 696 STANDARD. LOW POWER. KITS ARE FULLY SOCKETED. FULL DOC AND SCHEMATICS INCLUDED. 24 BIT ADDRESSING.

NEW! **\$59⁹⁵** BARE PC BOARD **\$219⁰⁰** 128K RAM KIT **\$139⁰⁰** 128 EPROM KIT

256K S-100 SOLID STATE DISK SIMULATOR!
WE CALL THIS BOARD THE "LIGHT-SPEED-100" BECAUSE IT OFFERS AN ASTOUNDING INCREASE IN YOUR COMPUTER'S PERFORMANCE WHEN COMPARED TO A MECHANICAL FLOPPY DISK DRIVE.

PRICE CUT!



- FEATURES:
- * 256K on board, using + 5V 64K DRAMS.
 - * Uses new Intel 8203-1 LSI Memory Controller.
 - * Requires only 4 Dip Switch Selectable I/O Ports.
 - * Runs on 8080 or Z80 S100 machines.
 - * Up to 8 LS-100 boards can be run together for 2 Meg. of On Line Solid State Disk Storage.
 - * Provisions for Battery back-up.
 - * Software to mate the LS-100 to your CP/M* 2.2 DOS is supplied.
 - * The LS-100 provides an increase in speed of up to 7 to 10 times on Disk Intensive Software.
 - * Compare our price! You could pay up to 3 times as much for similar boards.

BLANK PCB
(WITH CP/M* 2.2
PATCHES AND INSTALL
PROGRAM ON DISKETTE)
\$49⁹⁵
(8203-1 INTEL \$29.95)

\$129⁰⁰
(ADD \$50 FOR A&T)
#LS-100 (FULL 256K KIT)

ZRT-80

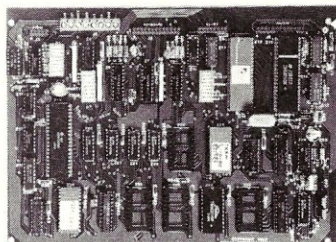
CRT TERMINAL BOARD!

A LOW COST Z-80 BASED SINGLE BOARD THAT ONLY NEEDS AN ASCII KEYBOARD, POWER SUPPLY, AND VIDEO MONITOR TO MAKE A COMPLETE CRT TERMINAL. USE AS A COMPUTER CONSOLE. OR WITH A MODEM FOR USE WITH ANY OF THE PHONE-LINE COMPUTER SERVICES.

FEATURES:

- * Uses a Z80A and 6845 CRT Controller for powerful video capabilities.
- * RS232 at 16 BAUD Rates from 75 to 19,200.
- * 24 x 80 standard format (60 Hz).
- * Optional formats from 24 x 80 (50 Hz) to 64 lines x 96 characters (60 Hz).
- * Higher density formats require up to 3 additional 2K x 8 6116 RAMS.
- * Uses N.S. INS 8250 BAUD Rate Gen. and USART combo IC.
- * 3 Terminal Emulation Modes which are Dip Switch selectable. These include the LSI-ADM3A, the Heath H-19, and the Beehive.
- * Composite or Split Video.
- * Any polarity of video or sync.
- * Inverse Video Capability.
- * Small Size: 6.5 x 9 inches.
- * Upper & lower case with descenders.
- * 7 x 9 Character Matrix.
- * Requires Par. ASCII keyboard.

FOR 8 IN. SOURCE DISK
(CP/M COMPATIBLE)
ADD \$10



\$89⁹⁵ A&T
#ZRT-80 ADD
\$50
(COMPLETE KIT, 2K VIDEO RAM)

BLANK PCB WITH 2716
CHAR. ROM. 2732 MON. ROM
\$49⁹⁵

SOURCE DISKETTE - ADD \$10
SET OF 2 CRYSTALS - ADD \$7.50

64K \$100 STATIC RAM

\$99⁰⁰
KIT

LOW POWER!

150 NS ADD \$10

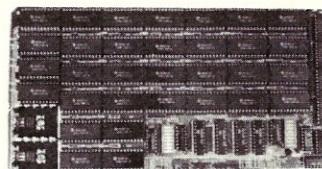
BLANK PC BOARD
WITH DOCUMENTATION
\$49.95

SUPPORT ICs + CAPS
\$17.50

FULL SOCKET SET
\$14.50

FULLY SUPPORTS THE
NEW IEEE 696 S100
STANDARD
(AS PROPOSED)

ASSEMBLED AND
TESTED ADD \$50



FEATURES: PRICE CUT!

- * Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
- * Fully supports IEEE 696 24 BIT Extended Addressing.
- * 64K draws only approximately 500 MA.
- * 200 NS RAMs are standard. (TOSHIBA makes TMM 2016s as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
- * SUPPORTS PHANTOM (BOTH LOWER 32K AND ENTIRE BOARD).
- * 2716 EPROMs may be installed in any of top 48K.
- * Any of the top 8K (E000 H AND ABOVE) may be disabled to provide windows to eliminate any possible conflicts with your system monitor, disk controller, etc.
- * Perfect for small systems since BOTH RAM and EPROM may co-exist on the same board.
- * BOARD may be partially populated as 56K.

PANASONIC

Green Screen - Video Monitors

25 MHZ. TYPICAL BANDWIDTH!!!

Brand New In The Box! 9-Inch Screen

#K-904B1 (Chassis #Y08A) Open Frame Style

\$29⁹⁵
EA.

GROUP SPECIAL:
4 for \$99⁰⁰
WITH DATA & SCHEMATIC

(USA SHIPPING: \$3. PER UNIT. CANADA: \$7. PER UNIT)

COMPUTER MANUFACTURER'S EXCESS. STILL IN ORIGINAL PANASONIC BOXES. THE CRT TUBE ALONE WOULD COST MORE THAN OUR PRICE FOR THE COMPLETE UNIT. FOR SPLIT VIDEO (TTL INPUTS) OPERATION, NOT COMPOSITE VIDEO. OPERATES FROM 12VDC AT 1 AMP. VERTICAL INPUT IS 49 TO 61 HZ. HORIZONTAL INPUT: 15,750 HZ ± 500 HZ. RESOLUTION IS 800 LINES AT CENTER 650 LINES AT CORNERS.

THE NEW 65/9028 VT ANSI VIDEO TERMINAL BOARD!

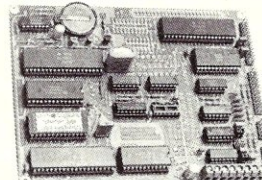
* FROM LINGER ENTERPRISES *

A second generation, low cost, high performance, mini sized, single board for making your own RS232 Video Terminal. Use as a computer console or with a MODEM for hook up to any of the telephone-line computer services.

FEATURES:

- * Uses the new SMC 9028 Video Controller Chip coupled with a 6502A CPU.
- * RS-232 at 16 Baud Rates from 50 to 19,200.
- * On board printer port!
- * 24 X 80 format (50/60 Hz).
- * For 15,750 Hz (Horiz.) monitors.
- * 3 Terminal Modes: H-19, ADM3A, and ANSI X 3.64-1979
- * Wide and thin-line graphics.
- * White characters on black background or reversed.
- * Character Attributes: De-Inten, Inverse, Underline and Blank.
- * Low Power: 5VDC @ .7A, ± 12VDC @ 20MA.
- * Mini size: 6.5 X 5 inches.
- * Composite or split video.
- * 5 X 8 Dot Matrix characters (U/L case) with descenders.
- * Answer back capability.
- * Battery backed up status memory.
- * For ASCII parallel keyboard.

MICRO SIZE!



\$99⁹⁵
(Full Kit)

ADD \$40 FOR A&T

SOURCE DISKETTE:
PC/XT FORMAT
5 1/4 IN. \$15

OPTIONAL EPROM FOR
PC/XT STYLE SERIAL
KEYBOARD: \$15

Digital Research Computers

P.O. BOX 381450 • DUNCANVILLE, TX 75138 • (214) 225-2309

TERMS: Add \$3.00 postage. Orders under \$15 add 75¢ handling. No C.O.D. We accept Visa and MasterCard. Tex. Res. add 5-1/8% Tax. Foreign orders (except Canada) add 20% P & H. Orders over \$50 add 85¢ for insurance.

Power to the PC!

by Stuart A. Jones

In my brief tenure as an IBM Compatible User Group Coordinator and general-purpose Personal Computer Maven, I have been surprised by how quickly my fellow users become dissatisfied with their computers' performance. A recurring topic at club meetings is how to wring the last millisecond out of a dBASE sort, a Lotus recalculation, or a Cobol compilation. As *Resident Expert* (of sorts), I usually get drawn into these discussions. This article represents a summary of the collective wisdom of various users, who have tried one or another of the gadgets and dodges described below in order to enhance the performance of their computer systems with varying degrees of success. IBM and third-party manufacturers have produced a dizzying array of products for PC enhancement. Many of these work as advertised, some better than advertised, and a few should be floated out to sea to feed silicon-loving marine organisms. My fellow users and I have sorted through these by trial and error. The discussion below will hopefully spare you, gentle reader, from the various trials and errors of our collective experience.

IS THIS UPGRADE NECESSARY?

Before embarking on a PC Enhancement Program, there are several questions you should ask yourself. First and foremost, you should determine whether you really *need* more performance. Apart from direct costs, there are potential problems with added unreliability (from either a flakey addition or an unexpected interaction between the add-on and existing hardware and software) as well as and potential incompatibility with current components. The *kid-in-a-candy-store* approach to enhancing PC's is a sure path to disaster. Therefore, a little *cost-benefit analysis* is in order.

There are a few general observations which can help you make the decision whether or not to upgrade:

1. If your pet PC is used for business, and you find yourself *hung up* doing necessary computer work in one of the ways described below, you *must* upgrade, since saving your valuable time will likely pay for the improvement in short order. In most major corporations, any purchase which pays for itself in one year is *mandatory*! This rule should be applied to PC's, as well

A Strategic Guide To Improving PC Performance

as everything else.

2. If your current frustration level with your PC's performance is such that you are often tempted to throw the bloody thing out the window, then an upgrade may well pay for itself by preserving your sanity.

3. If you find yourself frequently looking for excuses to use the PC/AT down the hall because you are weary of waiting for your PC to grind its way through the work, it's likely to be upgrade time.

Clearly, these criteria apply mainly to PC's which are being used for *serious* purposes. Unless your last name begins with *Rock* and ends with *feller* (in which case you can afford your own mainframe), it makes little sense to enhance a machine used mainly for fun and games. (Indeed, certain upgrades make game-playing impossible because of excessive execution speed or incompatibility with copy protection.) Obviously, also, if you are a *hacker*, whose joy in life is adding new stuff to his computer, this discussion doesn't apply. However, most PC users I know are using their machines for at least some work with economic impact, and therefore should pay attention to these issues.

WHERE'S THE HOLDUP?

If you decide your PC is indeed slowing you down intolerably, the next question is, what part of your computer system is the culprit? There are five common PC components which can cause performance problems, each of which causes a particular set of symptoms:

1. The CPU. If you are *CPU bound* you are spending too much time watching your PC *think* between keyboard inputs or disk accesses.

2. The disk. If you are *disk bound*, you spend too much time observing the little red lights on your disk drives and listening to grinding noises from disk-head movement.

3. The printer. You spend too much time waiting for hardcopy, not enough time computing.

4. The keyboard. At the end of each day, your fingers feel like they've run a marathon. However, be aware that a CPU-bound application may appear keyboard-bound if the computer spends too much time *thinking* between inputs.

5. The software. Your buddy across town performs the same size and type of spreadsheet analysis as you do, on a similarly-equipped PC, in half the time, using a different program.

For each of these hang-ups, there are specific cures, most of which are not worse than the disease. But before we examine solutions, there are two more issues which must be addressed: the size of your upgrade budget and how much technical support you have available.

The amount you are willing to spend will likely be determined by your initial cost-benefit analysis. If your application is critical, and the time-savings likely to be great, the sky may be the limit. Often, your budget will be set for you, hopefully by sympathetic financial executives (they tend to be heavy PC users, and will know whereof you speak). In smaller organizations, however, upgrade money may be tight, and your options therefore constrained.

The degree of technical support available may be critical in choosing your enhancements. Some upgrades can be managed by anyone capable of wielding a screwdriver; others require Peter Norton (or local equivalent thereof) standing at your elbow. Your available support may also determine the price you pay for improvements. If you have good in-house support, you can purchase almost anything you need by mail-order, knowing that the local experts can deal with potential problems. Otherwise, you may be better off buying from the nearest capable full-service dealer, paying a higher price, but being assured of assistance.

Now that we have explored these preliminaries, let us examine possible solutions for each of the five problem areas.

HELP FOR THE CPU BOUND USER

If you indeed need a PC which can compute faster, you should first ask yourself if you need a *new* machine altogether. At present, there are three

grades of PC mainframe available in the marketplace:

1. PC-class machines. These use an 8088 microprocessor chip, and run at a clock speed of 4.77 MHz. With a few exceptions, non-IBM examples tend to be 95%+ PC-compatible. Note that the IBM PC/XT, despite its hard disk, is a PC-class machine.

2. Intermediate-class machines. These typically use an 8086 or 80186 CPU chip, and run at 4.77 to 8 MHz. These tend to *think* 1.4 to 2.5 times as fast as a standard PC, and can still be very compatible, though some machines in this class (notably the Wang PC and Tandy 2000) are not very. Certain machines which use the 8088 chip and run at high clock speeds (up to 8 MHz, which gives 1.6 x PC performance) fit in this category. Some of these (notably the AT&T PC-6300 and certain *Turbo* clones) are bargains, costing slightly *less* than a standard PC.

3. AT-class machines. Using an 80286 microprocessor, these computers are typically only about 85% PC-compatible (though most business applications programs tend to run OK), but can compute 3 to 6 times as fast as a PC.

Upgrading your mainframe is technically simple, as long as you can be assured that the gadgets already plugged into your old PC will work with the new one. External gizmos—printers and modems, for example—are generally not a problem unless your current hardware uses non-standard cables. However, internal modems, memory expansion cards, and especially non-standard graphics, laboratory, and communications cards, may not operate properly in a different machine. Nonetheless, if you really need an AT-class machine, the cost of replacing existing add-ons may not be a significant deterrent.

If you are using an AT-class machine and are *still* CPU-bound, you should certainly ask yourself if your application(s) belong on a minicomputer instead! However, it may still be possible to enhance your computer's thinking speed by plugging in a clock crystal which makes it run at 8 or 10 MHz instead of the standard 6 MHz. (IBM's current version of ROM-BIOS precludes use of a higher-frequency clock crystal, but the clock crystal may work on earlier AT's, or AT-class machines from other manufacturers.)

If your calculations involve heavy use of *floating-point numbers* (e.g. spreadsheet, financial, scientific, or engineering applications), an 8087 or 80287 numerical-data-processing chip is a very desirable option as it is relatively cheap (\$100-300), easily installed, and causes

no software incompatibilities. However, your applications software *must* support use of the 8087/80287 in order for you to benefit from its installation. Also, be aware that the version and permitted clock speed of the numerical data processor chip must be matched to your computer. AT-class machines require the 80287; PC-class machines use the much cheaper 8087 or 8087-3; and intermediate-class machines which run at faster than 5 MHz require the 8087-2, whose price is likewise intermediate. Also, it pays to install a high-speed 80287 variant (a small circuit board that holds the chip plus a clock to drive it, all of which plugs into the 80287 socket) into a PC/AT, since the standard PC/AT drives the 80287 at a *slow* 3 MHz. MicroWay also produces this type of product; the one user I know who has one loves it.

If you need a CPU upgrade, and your budget is limited, a co-processor card may be your best option. This is a plug-in, PC-bus-compatible circuit board which holds a different and faster CPU chip (8086, 80186, or 80286), as well as special high-speed memory (usually 512 to 640 Kb) and support circuits. There are two varieties of PC-compatible co-processor cards. One requires that you remove the 8088 chip in your PC and plug in a header connector, in addition to installing the board in an available slot; the other just drops in to a slot. Boards using the *header* approach tend to be a bit more software compatible than the *drop ins*, but require some local technical assistance for installation. The *drop in* cards are easier to install, but require some software installation as well. However, the same software usually enables you to have your PC ignore the co-processor when you need 100% PC-compatibility. Prices for 8086- and 80186-based co-processor cards are dropping rapidly into the \$700 to 900 range. The 80286-based cards are still very expensive (typically more than \$2000) and therefore not as attractive. Common *header* cards are the MicroWay *Number Smasher* (which includes a high-speed 8087 chip) and the Univation card (which is similar to the *Number Smasher* and about the same price). The most common *drop in* card is the Orchid PC-Turbo Board, which is 80186-based, and about 90 to 95% PC-compatible. There are many other commercial products in this category, however, and new ones with better features are constantly being introduced.

HELP FOR THE DISK-BOUND USER

If you happen to be doing serious work with your PC, and are still using only diskettes for mass storage, RUN, don't walk, to the nearest appropriate supplier, and buy a hard-disk drive. Prices for Winchester hard drives have fallen so low, so fast, that it is ridiculous to be without

one. However, be aware of the hidden costs of a hard drive:

1. If you keep changeable data (e.g. inventories and correspondence) on the hard disk, you *must* establish a routine for copying this data to floppy disks (or use some sort of tape backup, if you can afford to) at frequent intervals. I have heard it said that the only difference between an expensive hard disk and a cheap one is that the expensive disk takes a little longer to break down! All Winchesters fail eventually, and when they do, all data on them is invariably lost. Some users I know keep only programs on their hard disk, and store all data on floppy disks.

2. Since hard-disk failure is inevitable, you must plan at some point for replacement costs. Some users buy two hard drives of the same size, and use one as a backup, not unreasonable, since a second drive typically costs less than a tape drive.

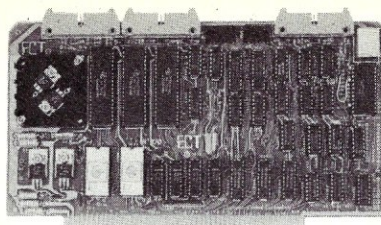
3. If you are not planning to buy a tape drive along with your hard disk, you should invest in either FastBack or Backup. These software products vastly improve on the pokey speed of the DOS *backup* command in transferring data to floppy disk. If backup is relatively painless, you will be less tempted to cheat (and potentially lose).

If you are using a PC or a clone (5 slots, 60-odd watt power supply), you should purchase an external hard disk drive, which typically costs about \$200 more, but which will not overburden (and possibly burn out) your computer's power supply. Even if you have a larger power supply, the external drive is easier to install (and replace when it breaks). Installing an internal drive may mean giving up one of your disk drives (making disk-copying painful), and requires some technical help. If you have an old IBM-PC with the 64 Kb motherboard (sold before February 1983), you may wish to try a ROM-BIOS upgrade (\$30 from IBM, if you can find one). This will enable you to start DOS from your hard disk. However, be aware that the ROM upgrade doesn't work with certain memory expansion boards and video-display cards. Having to *boot* from floppy disk is a minor inconvenience, but a significant one for less patient users.

If you need a hard disk, have data that requires high security, and/or need the ultimate in convenient backup, consider an Iomega Bernoulli Box. This product stores 10 or 20 Mb of data on 8" cartridges which are very rugged, and has excellent hard-disk performance specifications as well. The only problem with this excellent add-on is its price, over \$3000 list for a two-drive subsystem. However, that's not much more expensive than a premium-grade 40-Mb Winchester sub-

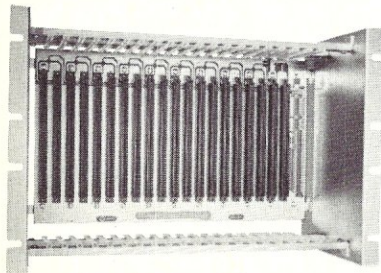
CUSTOM PRODUCTS

DESIGN • LAYOUT
MANUFACTURING

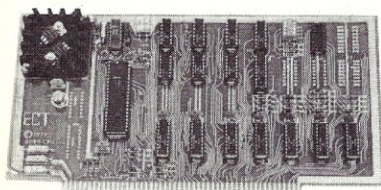


R 2I/O
ROM/RAM & I/O

S-100 PRODUCTS

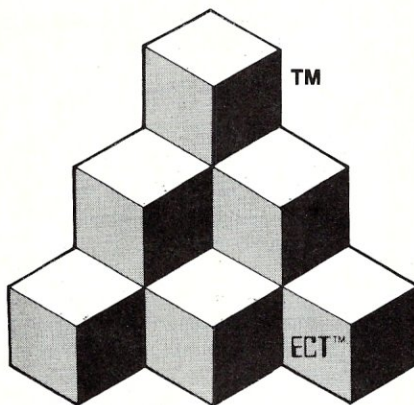


ECT-100-F
RACKMOUNT CARD CAGES



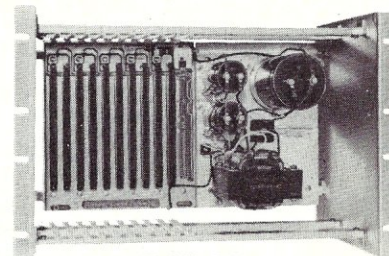
8080 CPU
CENTRAL PROCESSING UNITS

ECT™

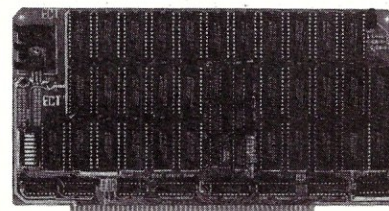


**BUILDING BLOCKS
FOR
MICROCOMPUTER SYSTEMS,
DEDICATED CONTROLLERS
AND TEST EQUIPMENT**

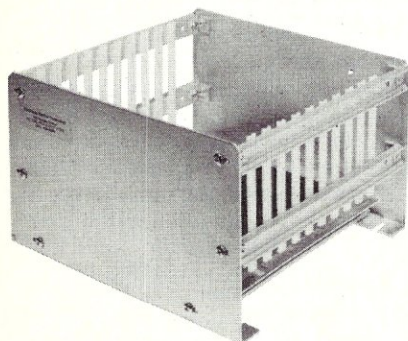
CARD CAGES, POWER SUPPLIES
MAINFRAMES, CPU'S, MEMORY
I/O, OEM VARIATIONS



RM-10
CARD CAGE & POWER SUPPLY



64K RAM
FULLY STATIC MEMORY



CCMB-10-F MIN
6, 10 OR 20 SLOT CARD CAGES

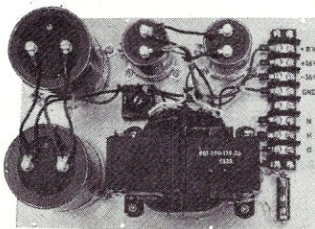


TT-10
TABLE TOP MAINFRAMES

ELECTRONIC CONTROL TECHNOLOGY, INC.

10 Cottage St., Berkeley Heights, NJ 07922 (201) 464-8086

**SPECIALIZING IN
QUALITY
MICRO COMPUTER
HARDWARE**



PS-30 A
POWER SUPPLIES

MULTIBUS® PRODUCTS

MULTIBUS IS A TRADEMARK OF INTEL CORP.

system with tape backup, and gives you unlimited storage (just add cartridges, about \$80 a piece). Definitely a great idea, if you can afford it!

Users who already have a hard disk and are still disk bound have two options: purchasing a higher-performance hard disk drive, or using one of the new high-capacity, memory-expansion boards to provide a large ramdisk. If your disk-binding problem involves sorting data bases, the latter possibility is both relatively inexpensive (under \$450 for 2 Mb from many sources) and simple to install and use (plug in the card and install the DOS-driver software). High-performance 5¼" hard disks (average access time is 40 msec. or less) are more expensive, but the data doesn't disappear when the power's shut off, like ramdisk. PC peripheral manufacturers have recently begun producing hard-disk controller cards using the SCSI (Small Computer Systems Interface) standard. Many different very high-performance 8" or 12" Winchester disk drives (average access time is less than 20 msec.) can therefore be connected to a PC, but at an expensive price (more than \$6000 for 80 Mb). These drives typically produce lots of heat and noise, and often are physically larger than the PC they're connected to.

HELP FOR THE PRINTER-BOUND USER

Very few users, in my experience, are truly printer-bound, in part because fast (160 cps or more) printers are so common, and also since the solutions discussed below are so inexpensive and widely known.

The two most common tactics used to avoid printer-binding are:

1. Buy a faster printer. The printer performance/price ratio has skyrocketed over the past three years. As an illustration, I just traded in a wide-carriage Epson MX-100 printer (November '82 vintage) for an FX-185 model from the same manufacturer. The new printer cost half as much, prints 2.5 times as fast, and has excellent near-letter-quality print built in as well. If you do a lot of printing and your printer is over 2 years old, you should at least check at the marketplace.
2. Employ a printer buffer. This may be done with either hardware or software. A hardware printer buffer is a small memory unit which plugs between your printer and the PC, and temporarily stores printout data until the printer is ready for it. A hardware buffer may be simulated by a small program which sets aside part of the PC's memory as temporary printout storage. Print-buffering programs (often called *spoolers*) usually stay resident in

BD Software, Inc., maker of the original
CP/M-80 C Language Development
System, knows

Time is precious

So the compilation, linkage and execution speeds of BDS C are the fastest available, even (especially!) on floppy-based systems. Just ask any user! With 15,000+ packages sold since 1979, there are *lots* of users . . .

New! Ed Ream's RED text editor has been integrated into the package, making BDS C a truly complete, self-contained C development system.

Powerful original features: CDB symbolic source-level debugger, fully customizable library and run-time package (for convenient ROM-ing of code), XMODEM-compatible telecommunications package, and other sample applications.

National C User's Group provides direct access to the wealth of public-domain software written in BDS C, including text editors and formatters, BBS's, assemblers, C compilers, games and much more.

Complete package price: \$150.
All soft-sectored disk formats, plus Apple CP/M, available off-the-shelf. Shipping: free, by UPS, within USA for *prepaid* orders. Canada: \$5. Other: \$25. VISA, MC, COD, rush orders accepted.

BD Software, Inc.

BD Software, Inc.
P O Box 2368
Cambridge MA 02238
617 • 576 • 3828



Mail

continued from page 11

editor, ZBASIC, and the relocatable Z80 assembler from Mitek as well as the PL/I-80 compiler. So far, I'm happy with them because they seem to be complete implementations of the languages (except PL/I-80 but it has enough). I also ordered the Z80 debugger from Mitek. From the documentation, it seems to be a full screen debugger that is patterned after some of the full screen debuggers for the PC type computers. Mitek sent me the wrong disk, so I had to send it back for replacement. Borland let me upgrade the version of Turbo Pascal (2.0) that I ran on the C-64 to a version that runs on the C-128 as part of their upgrade to 3.0 plan. Currently, PL/I-80 is my favorite language because of the extensive capabilities built into it. However, very few magazine articles seem to be written on it. Strangely, some versions of computer languages available in Commodore's native operating system seem to be better than what is available in CP/M. Superforth-64 has the most extensive library of any version of Forth that I've seen. PROMAL is the only high-level language compiler that I've seen that generates object code that runs close to assembly language speed. I find the full version of COMAL is an interesting language. It was available in the US on the C-64 before it was available on other computers and operating systems. There is a version of Pascal (Super Pascal) on the C-64 that even though it does not do in memory compiles, it corrects some of the short-comings of Turbo Pascal. It allows passing of functions and procedures as parameters (this is useful to me for numerical analysis), it has a built-in assembler so procedures can be written in assembly language instead of imbedded machine language, and PUT GET standard NEW DISPOSE READ WRITE and an unrestricted GOTO are all implemented. When I was debugging programs in batch mode on mainframe computers, I found memory dumps very helpful especially when turn around was every four hours. When interactive programming became available, using display or print statements in programs became a popular method of debugging. This may still take time on difficult programs. One C language program took me several days to find the problem using printf statements. Super Pascal has a useful option to provide a *post-mortem-dump* — a listing of the values of all the variables, a dump of the heap, and a trace-back of procedure calls and parameter stacks when a run time error occurs. I probably need to learn how to use a machine language debugger to track difficult high level language problems. Sometimes a dump is more useful than a trace.

The Commodore-128 will make an in-

continued on page 49

memory as *background* tasks, stealing CPU time to send data to the printer. The hardware buffer is preferable, since print-buffering software takes up memory space and CPU time, and may be incompatible with other programs. Also, the price of hardware buffers has fallen with the price of memory. A 64 Kb buffer (holding about 20 pages of printout data) can now be bought for less than \$150.

If your favorite PC applications involve a great deal of printing, and especially if other users in your organization use printers heavily, a laser printer may be very cost-effective. This type of printer is virtually silent, produces excellent letter-quality print (and in some cases graphics, too), prints 6-8 pages per minute, and is presently available for under \$3000 (graphics-capable printers cost much more, however). Laser printers can be shared, either via cable switches, or by using a low-cost PC-compatible as a printer-driver. (An expensive network is not necessary for printer-sharing. I personally prefer the *frisbee method* of print-serving: flip a floppy over to the print-PC.) Everyone I know who uses a laser printer is ECSTATIC, and the envy of us lesser mortals! A few laser-printer users are producing near-typeset quality newsletters, via special software which takes text files and formats them for the printer. Desktop publishing is quite a bit more expensive than plain laser-printing with typical hardware/software packages is in the \$12,000 range.

HELP FOR THE KEYBOARD BOUND USER

As mentioned earlier, it is uncommon for a PC-user to be truly keyboard-limited; complaints regarding slow keyboard response usually reflect a CPU-performance problem. However, there is a small but growing class of users who spend all day entering spreadsheet data, processing words, inputting inventory, thereby wearing their fingers out. There are a few products, all relatively inexpensive, which can make heavy keyboarding much easier.

The most obvious upgrade is a better keyboard. *Better*, in this case is relative; everybody has their own preference for keyboard *feel*, audible keystroke feedback, keyboard layout, etc. Word-processing users may consider learning how to use a keyboard with the Dvorak layout, after the inevitable adjustment period. Using such a keyboard reputedly improves throughput by 30 to 40%. If you want a Selectric layout and keyboard feel, no problem. Spreadsheet users can benefit from a keyboard with separate arrow and number keypads. Such replacement keyboards currently retail for less than \$200, and may be worth many times that in terms of operator comfort.

Spreadsheet users who frequently enter data into large templates may also benefit from one of the new keyboards with built-in pointing devices, which give mouse-like capabilities without taking your hands off the keys. (I particularly dislike *mus computeralis* (the common digital mouse) for keyboard work. Meeces (pl. mouse) were built for three-handed PC users, of whom I know none!) Units with built-in trackballs, joysticks and pointer-pads (like the well-known Koala pad) are commercially available in the \$250-400 range.

Anyone who enters the same, or similar, material repeatedly should be using a software keyboard enhancer, such as ProKey, SuperKey, or SmartKey. These programs assign multiple characters to a single keystroke, and can permit *fill in the blank* substitutions in an assignment. A keyboard enhancer does take up some memory, but relatively little. Also, incompatibilities with other programs are rare except for a few word-processing packages which take full control of the keyboard away from DOS. (Fortunately, none of the best-selling word processor programs do this.) Keyboard enhancers are inexpensive (under \$125), and oh-so effective.

HELP FOR THE SOFTWARE-BOUND USER

Being *hung up* by software performance is the mirror-image of being CPU-bound. Indeed, if you are wedded to a program which is known to be a *dog* in terms of slow execution, one of the CPU upgrades noted above will likely improve the situation. This is not as uncommon a problem as one might think. Cut-throat competition in *horizontal* software (e.g. word-processing, database, spreadsheet, programs useful for almost anyone in business) has imposed high standards for program-execution speed. However, many *vertical market* (e.g. for specific businesses) packages are still very inefficient. An application poorly coded in interpreted BASIC can make even an AT-class PC look sluggish. Often, the program with the *best* features for a specific situation is also the worst performer. As competition rises in the *vertical* market, program execution speed will improve, but this is cold comfort for today's user. If you are in such a vertical-package bind, best grit your teeth and upgrade your CPU.

One possible and easily addressed cause of poor software performance is repeated disk accesses for program overlays (parts of the program which have to be called from disk to memory to perform various functions). Expanding memory and using ramdisk to hold the overlay files can end this problem inexpensively. This method also works wonders for programs whose overlays are so big that they all

won't fit on a single diskette (noteable examples being DisplayWrite 2 and 3). Using a hard disk also works well, but imposes other costs (see above).

If your favorite spreadsheet or word-processor causes you to age visibly between recalculations or block moves, you should at least look at the market for alternative software. Bitter competition has caused prices to plummet, and performance/price ratios to skyrocket, even more so than for printers! You may also find that there is a new version of your favorite program, with upgrades available for a pittance, and much better performance. A classic example is the dramatic difference between Microsoft Word versions 1.0 and 2.0. The former was a veritable mutt; the latter, an industry leader. If you wish to switch manufacturers, it is likely that your existing data files will be at least partially transferable to the new program, though be sure to check this out thoroughly before buying! Also, be sure to include the expense of re-learning/re-training on the new software in your cost/benefit analysis.

If you are uncertain where your favorite package finishes in the performance sweepstakes, get a hold of an index to Software Digest and see if your program has figured in one of their excellent competitive reviews. The issue you need will likely be sitting somewhere in Corporate Computer Headquarters, or in the basement of someone like myself (yes, I subscribe). The review may not tell you if someone else's program will work better for your particular applications, but will let you know if your current package is really much slower than the competition.

The BOTTOM LINE

Though there are many ways to add power to a PC, in order to upgrade effectively, it is desirable to use the following strategy.

1. Identify what part(s) of your computer need improvement.
2. Determine if an enhancement is cost-effective.
3. Pick a solution in light of your available resources, budgetary and technical, and the potential costs and benefits of the proposed upgrade.

In my experience, individuals who take such a systematic approach are almost invariably successful in improving PC performance and the quality of their lives as PC users. (u)

COMPANIES MENTIONED IN THIS ARTICLE

MicroWay
Box 79
Kingston, MA 02364
(617) 646-7341

Desktop Publishers, Int'l.	
PC/PROFILER™ Turbo charge your software. Take three minutes to find out where your program is spending its time. In that amount of time PC/Profiler will provide you with invaluable information about which routines are slowing your program down. PC/Profiler can be turned on/off during program execution. Typically, execution speed goes up 50 to 500 percent once the offending functions are located and improved. Works with any compiled language system compatible with LINK.EXE or PLINK(tm)-86. \$150.	PC/MAKE™ End program recompilation busy work! Using an easily designed ASCII script MAKE calculates which modules need to be recreated based on file creation times. This product is a must for programmers developing with multiple modules. Command line flags allow forced recreation of a single file, all files, or groups of files named in a command file. A "no execute" flag allows previewing the actions MAKE will take. #defines with C-like arguments simplify MAKE script changes. Many examples and an instructive manual included. \$75.
<div style="display: flex; justify-content: space-between;"> Great offer! Amazing offer! </div> <p style="text-align: center;">PC/MAKE + PC/PROFILER: \$175 — \$50 savings</p>	
RUN/C--The C Interpreter™ Full K&R C interpreter with integrated full screen editor. Quick to learn and use. Ideal for learning C, mastering functions, working with interrupts, q&d debugging, testing out ideas. Immediate mode. PC BASIC-like graphics built in. 8087. \$120. Send \$10 for demo.	PS™ True proportional spacing supported for Diablo-compatible printers. What you see is what you get. Fast, fully menu-driven system with true footnotes, headers, footers, mailmerge. PS takes the drudgery out of creating printable mathematical formulas with on-screen preview of equations, forms, line graphics and text. Macro language built in. Thousands of users in the academic community. PS Scientific: \$495. WP version: \$350. Send \$10 for demo. This ad Laserset(tm) with PS.
RUN/C Professional All the above features plus Loadable Libraries™ which provides a great interpreted environment for any .LIB or .OBJ routine created with the Lattice large model C compiler. Quickly create loadable versions of the Halo™ C-Food™ C-Utility™ functions you use. Load and unload multiple libraries in less than a minute. Complete C source-level debugging including multiple breakpoints, single stepping, etc. Interrupt program execution and determine/change the value of any variable. \$250.	Desktop Publishers (tm) PDS Three megabytes of the finest public domain software utilities and word processing aids. many gems. \$49.
<div style="display: flex; justify-content: space-between;"> <div> Desktop Publishers 151 First Ave. New York, NY 10003 (212) 473-2309 </div> <div style="border: 1px solid black; padding: 5px;"> All products carry a 30-day money back guarantee. </div> </div>	
<small>Prices guaranteed through May, 1986. Shipping and applicable taxes not included. General requirements: All products require MS-DOS (PC DOS) 2.0 or greater. PC/Dvorak! requires IBM PC or XT. EC requires PC or compatible. PS requires PC, Victor or Apricot. RUN/C is available for PC compatibles or MS-DOS systems with ANSI.SYS. Trademarks: Desktop Publishers, RUN/C -- The C Interpreter, Loadable Libraries, PC/Profiler, PC/MAKE, PC/Dvorak!, Laserset and Age of Reason are trademarks of Age of Reason Co.; PS TM Scroll Systems, Inc.; EC TM C Source. Halo TM Media Cybernetics; C Utility Library TM Essential Software; Lattice R and C-Food TM Lattice, Inc. PLINK-86 TM PCPC. MS-DOS TM Microsoft. UNIX TM AT&T. IBM R International Business Machines, Inc.</small>	
Mastercard / Visa / Eurocard	

Univation
1037 No. Fair Oaks Ave.
Sunnyvale, CA 94089
(408) 745-0180

Orchid Technology
47790 Westinghouse Dr.
Fremont, CA 94539
(415)490-8586

Backup
Software Integration Inc
9800 So. Sepulveda Blvd. #214
Los Angeles, CA 90045
(213) 776-3406

Koala Technologies Corp.
3100 Patrick Henry Dr.
Santa Clara, CA 95052

ProKey
RoseSoft
4710 University Way N.E.
Seattle, WA 98105
(206) 524-2350

Super Key
Borland International
4585 Scotts Valley Dr.
Scotts Valley, CA 95066
(408) 438-8400

SmartKey
Heritage Software Inc.
2130 S. Vermont Ave.
Los Angeles, CA 90007
(213)737-7252

Creating A Copy Protected Program

by Edwin Thall

Generate a diskette file That cannot be duplicated By MS/PC-DOS

You've probably noticed that copy protected programs work properly when executed from the original diskette, but fail when you attempt to run a copied version. Popular methods for preparing protected diskettes include unorthodox formatting, physical marking of the surface, and altering rotation speed. In this article, I describe key parameters for generating a diskette sector which cannot be copied by PC-DOS. I also demonstrate the step-by-step formation of a copy protected program. All examples and displays cited in this paper pertain to the DOS 3.00 360K format.

The read only memory (ROM) built into IBM PC's and compatibles store the routines needed to operate the various peripherals, such as the diskette drives. These routines are referred to as the Basic Input/Output System (BIOS). The ROM-BIOS is a permanent part of the computer hardware and cannot be altered.

MS/PC-DOS is manufactured by Microsoft, Inc. and is not part of the hardware. DOS provides numerous services such as creating new files, deleting old files, copying files, etc. Most of us rely on DOS to carry out these tedious chores and seldom, if ever, access the BIOS directly. Our programs call a DOS routine which in turn, calls a ROM-BIOS routine. However, DOS was created by mortals and has limitations. The changing of a key parameter can render MS/PC-DOS unserviceable.

DISKETTE PARAMETER TABLE

Software interrupt instructions enable programs to access ROM-BIOS routines. The ROM addresses of these routines are stored in the computer's first 1024 memory locations. This part of memory, known as the interrupt vector table, is loaded from the system diskette during the start-up. You may display the first 128 bytes of this table with the Debug utility. From DOS, type:

```
A)DEBUG
-D 0:0
```

```
0000:0000 5A 50 1F 01 ED 01 70 00-C3 E2 00 F0 ED 01 70 00
0000:0010 ED 01 70 00 54 FF 00 F0-47 FF 00 F0 47 FF 00 F0
0000:0020 A5 FE 00 F0 87 E9 00 F0-DD E6 00 F0 DD E6 00 F0
0000:0030 DD E6 00 F0 DD E6 00 F0-57 EF 00 F0 ED 01 70 00
0000:0040 65 F0 00 F0 4D F8 00 F0-41 F8 00 F0 54 09 70 00
0000:0050 39 E7 00 F0 59 F8 00 F0-2E E8 00 F0 D2 EF 00 F0
0000:0060 00 00 00 F6 75 0A 70 00-6E FE 00 F0 E7 01 70 00
0000:0070 53 FF 00 F0 A4 F0 00 F0-22 05 00 00 00 00 F0
```

Each interrupt is assigned four bytes pointing to the address of its routine. The four bytes are given in reverse order. For example, the first interrupt points to address 011F:505AH (H = hexadec-

imal system). Interrupt 30 (INT 1EH) does not point to a ROM-BIOS routine, but to a table of eleven consecutive bytes. This table, known as the diskette parameter table (DPT), defines how information is arranged on the diskette and regulates the operation of the diskette drives. The original DPT is stored in the ROM at address F000:EFC7H. You can look at this table with:

```
-D F000:EFC7,EFD1
F000:EFC7 CF 02 25 02 08 2A FF 50 F6 19 04
```

Starting with DOS 1.10, IBM modified this table to improve the performance of the diskette drives and the ROM table was replaced with one in the random access memory. The interrupt vector table reveals that INT 1EH (offsets 0078H-007BH) begins at address 0000:0522H. You can display the modified DPT with:

```
-D 0:522,52C
0000:0522 DF 02 25 02 09 2A FF 50 F6 0F 02
```

A description of the eleven diskette parameters is provided in Table 1. Offsets 522H, 523H, 524H, 52BH, and 52CH are required for the operation of the diskette drives and should not be changed. The other parameters define the diskette format and can be modified. The default values for these parameters are:

512	bytes per sector
9	sectors per track
42	bytes between sectors
80	bytes between ID and data fields
246	format data fill

The format data fill parameter specifies the binary pattern written to the data field. The decimal value (246) corresponds to F6H and 512 bytes of 11110110 are initially stored in formatted sectors.

THE CHRN

The 5¼" floppy is organized into 40 concentric tracks per side. The 360K format divides each track into 9 sectors. There are two major components to a sector: the identification (ID) field and the data field. Data is stored in the data field and is of course, the most important part of the sector. The ID field stores the information needed to locate sectors. The diskette organization by track and sector is illustrated in Figure 1.

Before a diskette may be used, each track must be individually formatted. A vital part of a sector's ID field, the CHRN, is written during this operation. The CHRN is a four-byte field representing the following:

C	Track number	(00-27H)
H	Head number	(00 = side 0, 01 = side 1)
R	Sector number	(01-09)
N	Sector length	(00 = 128, 01 = 256, 02 = 512, 03 = 1024)

For example, the CHRN for the sixth sector on track 09, head 00, and holding 512 bytes of data is 09000602. The standard sector length is 512 bytes (N = 02). While DOS can only locate sectors of this size, the BIOS can handle all allowed sector lengths. For protected diskettes, various combinations of the CHRN may be used.

DISKETTE INPUT/OUTPUT OPERATIONS

You can read, write, and format sectors at the BIOS level by means of interrupt 19 (INT 13H). The registers required for INT 13H operations are described in Table 2.

Table 1
Description of Diskette Parameters

Offset	Parameter	Function
522H	step rate time	controls rate head moves from track to track
523H	head load time	wait time before read/write
524H	motor wait time	delay until motor off
525H	bytes per sector	(00 = 128, 01 = 256, 02 = 512, 03 = 1024)
526H	end of track	number of sectors per track
527H	gap length	bytes separating sectors
528H	data length	used when sector length not specified
529H	gap length for format	bytes separating ID and data fields
52AH	fill byte for format	value stored in data field after format
52BH	head settle time	wait settle time before read/write
52CH	motor start time	delay until motor off

Table 3
BIOS INT 13H Error Codes

00	no error
01	bad command
02	address not found
03	diskette write protected
04	sector not found
08	direct memory access error
10	CRC error
20	disk controller error
40	seek error
80	attachment failed to respond

When a BIOS INT 13H operation fails, the diskette controller is disabled and must be reset for the next operation. The two instructions for reset are:

```
MOV AH,00
INT 13H
```

DOS detects and handles all hardware-related disk errors, but BIOS expects you to perform this job. The status of a BIOS INT 13H operation is returned to the carry flag (success or failure) and the AH register (cause of error). The carry flag is set to zero (NC) for a successful operation. The error codes returned to the AH register are listed in Table 3. You may encounter a read operation giving an error even though there is nothing wrong. This condition arises because the disk drive motor takes about a half second to reach full speed, and BIOS does not wait between starting the motor and issuing the read command to the disk interface. Therefore, the hardware may try to read from the disk before it is rotating at the proper speed. The IBM Technical Reference Manual suggests that "three retries are required on reads to ensure the problem is not due to motor start-up."

IRREGULAR SECTOR 08000101

Normal diskette sectors are 512 bytes in length and numbered in sequence. The CHRN's for the nine sectors located on track 08/head 00 are:

```
08000102
08000202
08000302
08000402
08000502
08000602
08000702
08000802
08000902
```

Irregular sectors deviate from this pattern and there are various ways to represent them. One approach is to number the sectors out of sequence (08000102, 08000702, 08000502, 08000202, etc.). Another method is to repeat CHRN combinations (08000102, 08000202, 08000202, 8000202, 08000502, 08000602, etc.).

Table 2
Registers Required for INT 13H Operations

Register	Function	Parameters
AH	Diskette Operation	(00-05*)
AL	Number of sectors	(01-09)
CH	Track number	(00-27H)
CL	Sector number	(01-09)
DH	Head number	(00=side 0,01=side 1)
DL	Drive number	(00=drive A,01=drive B)
ES:BX	Address of buffer	(ES=segment,BX=offset)

* 00	Reset disk system
01	Read status of system
02	Read sectors into memory
03	Write sectors to diskette
04	Verify sectors
05	Format track

Irregular sectors confuse DOS and the operating system has difficulty locating them.

Another technique for generating irregular sectors is to change the N (of CHRN) to 00, 01, or 03. Whereas DOS always uses N=02 as the sector length, the BIOS assumes the value stored in the diskette parameter table.

The development of my copy protection scheme begins with the formation of an irregular sector. I suggest that you follow with an experimental diskette (360K format) containing only the operating system and Debug.

I have arbitrarily decided to create irregular sector 08000101. The first sector on track 08/head 00 will be assigned a data length of 256 bytes (N=01). The assembly language program to format this irregular sector, along with eight regular sectors, is provided in Figure 2.

I begin the Format program by resetting the diskette controller. Next, the registers are loaded to specify the operation, track, head, and drive. The BX register points to the series of the nine CHRN entries. There must be one entry for every sector formatted. Note the CHRN for the first sector on track 08 (08000101). The N parameter designates it an irregular sector. Use the assemble command (A 100H) to enter this program (do not include comments). After you type A 100H, the system will display DS:0100H and wait for you to enter the first instruction (MOV AH,00). When you are finished entering the Format program, hit (ENTER) twice to exit the assemble mode.

The Format program is stored in memory and is ready for execution. Before running the program, change the fill byte parameter from F6H to AAH. The contents of the irregular sector will be easily recognized by a long string of AA's. To alter the fill parameter, type:

```
-E 0:52A
0000:052A F6.AA
```

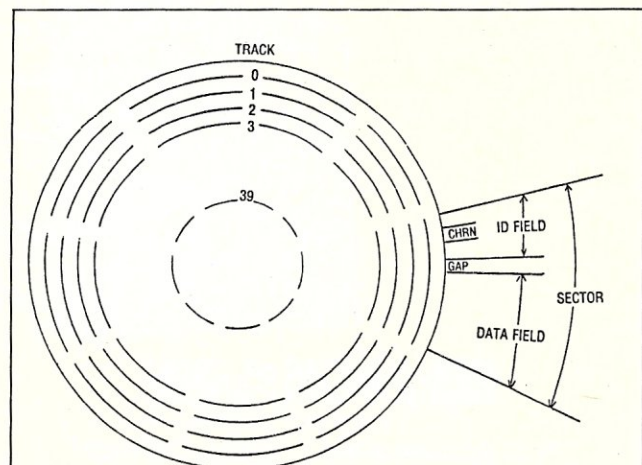


Figure 1. Diskette organization by track and sector (360K format)

DRIVE INN

Enclosure &
power supplies
for
**FLOPPY,
WINCHESTER,
TAPE DRIVES,
SINGLE BOARD
COMPUTERS
& S-100 SYSTEMS**

8 inch
5 inch
3 inch

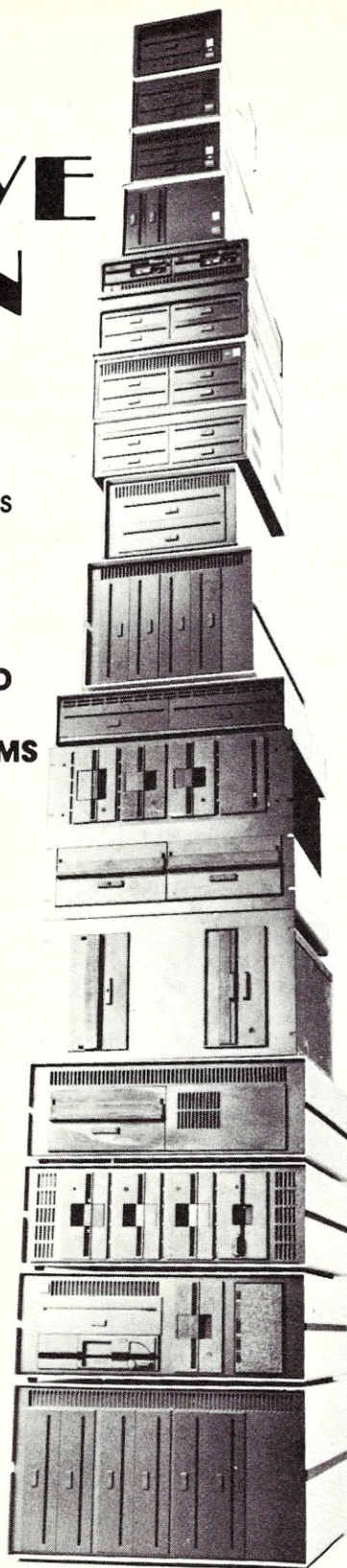
Call or write
for free
catalogs &
application
assistance

INTEGRAND

RESEARCH CORPORATION

8620 Roosevelt Ave. • Visalia, CA 93291
209/651-1203

We accept BankAmericard/Visa
and MasterCharge



```
DS:0100  MOV  AH,00      ;RESET DISK CONTROLLER
DS:0102  INT  13         ;BIOS DISK I/O
DS:0104  MOV  AH,05      ;FORMAT TRACK
DS:0106  MOV  AL,09      ;9 SECTORS
DS:0108  MOV  CH,08      ;STARTING AT TRACK 08
DS:010A  MOV  CL,01      ;      "      SECTOR 01
DS:010C  MOV  DH,00      ;      "      HEAD 00
DS:010E  MOV  DL,00      ;      "      DRIVE A
DS:0110  MOV  BX,0117    ;POINT TO CHRN
DS:0113  INT  13         ;BIOS DISK I/O
DS:0115  INT  20         ;RETURN TO DOS
DS:0117  DB   08 00 01 01 ;CHRN ENTRIES
DS:011B  DB   08 00 02 02
DS:011F  DB   08 00 03 02
DS:0123  DB   08 00 04 02
DS:0127  DB   08 00 05 02
DS:012B  DB   08 00 06 02
DS:012F  DB   08 00 07 02
DS:0133  DB   08 00 08 02
DS:0137  DB   08 00 09 02
```

Figure 2. Format program (creates sector 08000101)

Execute the Format program with the diskette in drive A.

-G

Sector 08000101 cannot be read by DOS. This sector is logical sector number 90H and if you attempt to read with the conventional Debug "L" command, the following error is returned:

-L 400 0 90 1

Disk error reading drive A

You must invoke a BIOS INT 13H operation to read sector 08000101. Change the Format program to a Read program with:

```
-E 105
DS:0105  05.02
-E 107
DS:0107  09.01
-E 111
DS:0111  17.00 01.08
```

The Read program stores the contents of sector 08000101 into a buffer area beginning at offset 0800H. You may display the Read program with the unassemble command (U 100H). In my display (Figure 3), I have included comments next to the three modified instructions.

Since irregular sector 08000101 was assigned 01 for its N parameter, the sector length parameter in the DPT must be set at this value before you attempt to run the Read program.

```
-E 0:525
0000:0525 02.01
```

If you wish to check the status of this read operation, you must do it prior to the program's return to DOS. You may run the program using G 115H in order to stop execution before control is returned to DOS (INT 20H). When offset 115H is reached, the program stops and the registers/flags are displayed.

-G 115

```
AX=1000 BX=0800 CX=0801 DX=0000 SP=FFFF BP=0000 SI=0000 DI=0000
DS=???? ES=???? SS=???? CS=???? IP=0115 NV UP EI NG NZ AC PE CY
DS:0115 CD20      INT      20
```

The error (AH=10) indicates that the read operation failed because of a cyclic redundancy check (CRC). This type of error occurs when the number of bytes written to a sector does not agree with the number read. The sector length parameter in the diskette

```
DS:0100  B400  MOV  AH,00
DS:0102  CD13  INT  13
DS:0104  B402  MOV  AH,02      ;READ OPERATION
DS:0106  B001  MOV  AL,01      ;ONE SECTOR
DS:0108  B508  MOV  CH,08
DS:010A  B101  MOV  CL,01
DS:010C  B600  MOV  DH,00
DS:010E  B200  MOV  DL,00
DS:0110  BB0008 MOV  BX,0800    ;BUFFER AREA
DS:0113  CD13  INT  13
DS:0115  CD20  INT  20
```

Figure 3. Read program (reads sector 08000101)

FOR THE Z-100

Conforms to
IEEE -696 (S-100)

A low-cost high-performance
memory board from
Software Wizardry, Inc.

- Up to 1 megabyte on a single card with parity error checking
- On-board clock/calendar with software
- The only memory card not manufactured by ZDS that is specifically designed for the Zenith Z-100
- Emulates the Z-205 registers for complete present and future Z-100 compatability
- Runs at 8 MHz with no wait states (when loaded with 150 nanosecond chips)
- Ramdrive software included
- Supports 24 bit addressing

List price: **\$395 (unpopulated)**
\$520 (with 1 megabyte)



Software Wizardry, Inc.
1106 First Capitol Drive
St. Charles, MO 63301
(314) 724-1738

```
DS:0100 B400    MOV    AH,00
DS:0102 CD13    INT     13
DS:0104 B402    MOV    AH,03    ;WRITE OPERATION
DS:0106 B001    MOV    AL,01
DS:0108 B508    MOV    CH,08
DS:010A B101    MOV    CL,01
DS:010C B600    MOV    DH,00
DS:010E B200    MOV    DL,00
DS:0110 BB0008  MOV    BX,0800    ;BUFFER AREA
DS:0113 CD13    INT     13
DS:0115 CD20    INT     20
DS:0800 MOV     DX,0809    ;MESSAGE OFFSET
DS:0803 MOV     AH,09      ;PRINT STRING
DS:0805 INT     21         ;CALL DOS
DS:0807 INT     20         ;RETURN TO DOS
DS:0809 DB      "THE IRREGULAR SECTOR WAS READ $"
```

Figure 4. Write program (writes 256 bytes to sector 08000101)

parameter table specified 512 bytes to be written during the format of the irregular sector but the read operation requested 256 bytes. You can display the contents of the buffer area with:

-D 800,8FF

You should be looking at 256 consecutive bytes of AAH. The irregular sector was read and an error was returned to indicate a discrepancy between the number of bytes read (256) versus the number written (512).

WRITING TO THE IRREGULAR SECTOR

For our copy protection scheme, a short program is written to the irregular sector. Convert the Read program (AH=02) to a Write program (AH=03) with:

-E 105
DS:0105 02.03

The routine to print "THE IRREGULAR SECTOR WAS READ" will be stored in sector 08000101. Enter the program in Figure 4 beginning with offset 0800H.

-A 800 DS:0800
MOV DX,0809

The 256 bytes (offsets 0800H-08FFH) are written to sector 08000101 when you run the program. Remember to reset the instruction pointer register (IP=0115H) to 100H before executing.

-RIP
IP 0115
:100
-G

Now attempt to read sector 08000101. You should be able to do so without the dreaded CRC error. Convert the current program in memory from a write (AH=03) to a read (AH=02) and then run.

-E 105
DS:0105 03.02
-G 115

AX=0000 BX=0800 CX=0801 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=???? ES=???? SS=???? CS=???? IP=0115 NV UP EI NG NZ AC PE NC
DS:0115 CD20 INT 20

The AH register (00) indicates that the read operation worked. Run the program again if an error was returned in the AH register. Restore the sector length parameter to the standard value (N=02). If you should quit Debug without making this adjustment, you will have to reset the computer to access any files on the diskette!

-E 0:525
0000:0525 01.02

MODIFY FAT

Irregular sectors may cause hardware errors and DOS must be notified when one exists. The FAT (file allocation table) stores sector access information. It is essential to modify the FAT for our copy protection scheme. The irregular sector is logical sector


```

DS:0000 FD FF FF 03 40 00 05 60-00 07 80 00 09 A0 00 FF
DS:0010 CF 00 0D E0 00 0F 00 01-11 20 01 13 40 01 15 60
DS:0020 01 17 80 01 19 A0 01 1B-C0 01 1D E0 01 1F 00 02
DS:0030 21 20 02 23 40 02 25 60-02 FF 8F 02 29 A0 02 2B
DS:0040 C0 02 2D E0 02 2F 00 03-31 20 03 33 40 03 35 60
DS:0050 03 37 80 03 39 A0 03 3B-C0 03 FF 0F 00 00 00 00
DS:0060 00 00 00 00 00 00 F7 0F-00 00 00 00 00 00 00
DS:0070 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

```

Figure 5. The first 128 bytes of the modified FAT

number 90H or 144. The cluster number in the FAT may be determined from the logical sector number (LSN) by:

Cluster number = $.5(\text{LSN} - 12) + 2 = .5(144 - 12) + 2 = 68$

LSN 144 corresponds to cluster 68 in the FAT. A cluster entry of FF7H is the code to inform DOS that sectors of the cluster are bad and should not be allocated. Offsets 0066H and 0067H in the FAT point to cluster 68. To modify the FAT, place the diskette in drive A and type:

```

-L 0011
-E 66
DS:0066 00.F7 00.0F
-W 0011
-L 0011
-D 0

```

The modified FAT (Figure 5) now contains F7H and 0FH at offsets 0066H and 0067H, respectively.

THE COPY PROTECTED PROGRAM

The name of the copy protected program is CPROTECT.COM. The function of this program is to execute the routine written to sector 08000101. CPROTECT.COM was

```

CSEG      SEGMENT
          ASSUME CS:CSEG
;SET DISKETTE PARAMETER TO 256 BYTES PER SECTOR
PUSH DS   ;SAVE ORIGINAL DS
MOV AX,0  ;DS FOR DISKETTE PARAMETER
MOV DS,AX ;SET DS=0
MOV BX,525H ;SECTOR SIZE OFFSET
MOV AL,1   ;256 SECTOR SIZE CODE
MOV [BX],AL ;SET SECTOR SIZE TO 256
POP DS     ;RESTORE ORIGINAL DS
;BIOS ROUTINE TO READ SECTOR 08000101
MOV CX,3   ;TRY 3 READS
AGAIN: PUSH CX ;SAVE COUNT
MOV AH,0   ;DISK RESET
INT 13H    ;BIOS DISK I/O
MOV AH,2   ;READ DISK
MOV AL,1   ;READ ONE SECTOR
MOV CH,8   ;FROM TRACK 8
MOV CL,1   ; " SECTOR 1
MOV DH,0   ; " HEAD 0
MOV DL,0   ; " DRIVE A
MOV BX,800H ;BUFFER AREA
INT 13H    ;BIOS DISK I/O
;ATTEMPT 3 TIMES IF READ FAILS
CMP AH,0   ;DID READ FAIL ?
JZ PASSED  ;NO--READ PASSED
POP CX     ;YES--POP READ COUNT
LOOP AGAIN ;TRY READ AGAIN
MOV CL,0FFH ;SET FAIL
;SET DISKETTE PARAMETER TO 512 BYTES PER SECTOR
PASSED: PUSH DS ;SAVE ORIGINAL DS
MOV AX,0  ;DS FOR DISKETTE PARAMETER
MOV DS,AX ;SET DS = 0
MOV BX,525H ;SECTOR SIZE OFFSET
MOV AL,2   ;512 SECTOR SIZE CODE
MOV [BX],AL ;SET SECTOR SIZE TO 512
POP DS     ;RESTORE ORIGINAL DS
;WAS THE IRREGULAR SECTOR READ ?
CMP CL,0FFH ;DID READ FAIL ?
JZ PRINTIM  ;YES--PRINT FAILED MESSAGE
MOV AX,800H ;NO--PRINT SUCCESS MESSAGE
JMP AX      ;AT OFFSET 0800H
;PRINT FAILED MESSAGE
PRINTIM: MOV DX,OFFSET MESS
MOV AH,9  ;PRINT STRING
INT 21H  ;CALL DOS
INT 20H  ;RETURN TO DOS
;STORE FAILED MESSAGE
MESS DB 0DH,0AH,'PROGRAM FAILED $'
SEG ENDS
END

```

Figure 6. Assembler code (CPROTECT.ASM)

assembled from the source code (CPROTECT.ASM) listed in Figure 6.

CPROTECT.COM changes the sector length parameter from 512 to 256 bytes. Next, the BIOS read routine stores the contents of sector 08000101 in a buffer area (offset 0800H). The AH register is compared to 00 to determine whether the read operation was successful. If three read operations fail, the CX register is set to FFH. The sector length parameter is restored to 512 bytes before control is returned to DOS. The message "PROGRAM FAILED" is displayed when three read operations fail (CX = FFH). Otherwise, the routine stored in sector 08000101 is executed and the message "THE IRREGULAR SECTOR WAS READ" is displayed.

I wrote CPROTECT.ASM and compiled it with the IBM Macro Assembler to form CPROTECT.OBJ. The object file was linked and converted to CPROTECT.COM.

SUMMARY

The copy protected program is complete. You can run CPROTECT.COM from DOS with:

A) CPROTECT

The program will display the message "THE IRREGULAR SECTOR WAS READ" when executed from the original diskette in drive A. You may copy CPROTECT.COM to a second diskette, but the program is destined to fail because the irregular sector was not copied. If you attempt to copy the entire diskette with the DISKCOPY command, the following messages are displayed:

```

Unrecoverable read error on source
Track 08, side 0
Target diskette may be unusable

```

(M)

Dr. Edwin Thall is Associate Professor of Chemistry at The Wayne General and Technical College of The University of Akron. He teaches chemistry and computer programming.

..... I S I S ■ ■

CP/M ➡ ➡ MS-DOS

COMPLETE SOURCE CODE INCLUDED

ICXPDS: eXchanger now supports the 5¼" iPDS format. Manipulation of ISIS-II files using your computer system was never easier.

ICXMDS: Same as ICXPDS, but for MDS 8" systems.

IMXPDS: Reads/Writes 5" iPDS disks on PC's and AT's.

TELEPLUS: Enhanced serial file transfer program for CP/M, ISIS, or MS-DOS.


ISE: Emulator gives the CP/M and MS-DOS user access to all the ISIS-II languages and utilities.

ACCEL 8/16: CP/M-80 emulator for MS-DOS. Enables PC's to run ISE. (no source code)

\$89 each
\$250 any 3

ZAS Development Package:
Z-8 / Z-8000 Assembler for CP/M and MS-DOS.

Request catalog of products

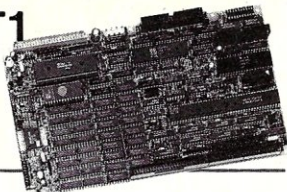


Copyrights: CP/M Digital Research, Inc.
ISIS-II and iPDS Intel Corp. MSDOS Microsoft

Western Wares 303-327-4898
Box C • Norwood, CO 81423

MSC-LAT1

\$649

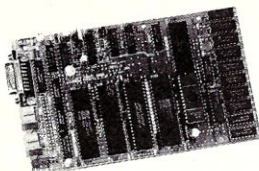


KAYPRO™

users can share the advantage to LAT1. Just take off your main KAYPRO board and put LAT1-K into your cabinet.

All advantage of LAT1 is yours now!

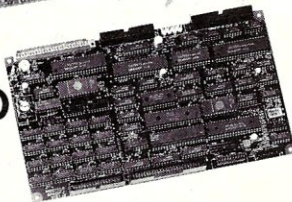
\$674



MSC-MTC

MSC-ICO

\$499



CP/M plus is a registered trademark of Digital Research Inc.
Z80 is a registered trademark of Zilog Inc.
Turbo Dos is a registered trademark of Software2000 Inc.
Mountain Side Computer and ZENET are trademarks of Southern Pacific Limited

Distributors

England-Quanta systems 01-253-8423
Denmark-Danbit 03-662020
Finland-BB.Soft 90-692-6297
India-Betamatix PVT Ltd. 0812-71989
Australia-LAMRON PTY. Ltd. 02-808-3666

Manufacturer and international distributor

SOUTHERN PACIFIC LIMITED

Sanwa Bldg., 2-16-20 Minamisaikai, Nishi, Yokohama, JAPAN 220
Phone: 045-314-9514 Telex: 3822320 SPACIF J
Advanced single board computer technology company

ZENET NETWORK through twist pair

- 6Mhz HD64B180 (Z80 upward compatible) 512K byte on board (256K installed, 384K RAMDISK)
- LAN:ZENET port 800K baud CSMA CD twist pair bus type upto 500 meters HDLC
- Floppy: 3.5, 5 and 8 inch, d/s density, d/s sided and d/s track automatic density/format checking
- Hard disk: SCSI interface on board
- Video: 80 X 24 characters (color) and 640 X 200 pixels color graphic 128K byte video RAM character set is downloaded from disk
- Timer: battery back up calendar

- Serial: RS232C X 2 and TTL X 1
- Pararell: centronics type, 16 bit TTL, 7/8 bit keyboard port (32 characters FIFO)
- O.S.: Turbo Dos, MP/M (multiuser) banked CP/M plus (single user)
- Size: 10 X 6 inch 4 layered
- Assembled and tested
- BIOS source code available
- Completely faster than other Z80SBC

MSC-PCX

8088 expansion card for LAT1 soon available

WORLD SMALLEST COMPUTER

- Full personal CP/M system in palm: 4mhz Z80 256K RAM (128K RAM DISK)
- Serial: RS232C X 2 automatic baud rate checking
- Pararell: centronics type printer port
- Floppy: 3.5 inch micro floppy disk drive 800K byte (option 5, 3.5 inch drive d/s sided d/s track automatic density checking)

- O.S.: CP/M plus bank version
- BIOS source code available
- Completely faster than other Z80SBC

MSC-MTC/P

Full assembled pcb of MTC
Under \$189 in OEM quantity

\$299

Full featured CP/M plus system

- Z80 4mhz 128K Byte RAM Floppy: 3.5, 5 and 8 inch d/s density, d/s sided and d/s track upto 4 disk drives Automatic density/format check
- Serial: RS232C X 2
- Pararell: Centronics type, 16 bits I/O, 7/8 bit keyboard port
- Timer: battery back up calendar
- Video: 80 X 24 high speed CRT controller
- O.S.: CP/M plus bank version included
- Size: 10 X 6 inch 4 layered

- BIOS source code available
- DRI CP/M plus manual \$50
- New word word processor program for MSC-ICO ADD \$50
- Completely faster than other Z80SBC

MSC-HCS

Expansion card for ICO
RAM disk (upto 2M byte) and SCSI hard disk interface card for ICO with installation program

\$199

USA distributor

SOUTHERN PACIFIC COMPUTER PRODUCTS U.S.A., INC.

21 Altarinda Rd. Orinda, CA 94563
Phone: 415-253-1270

Dealer and distributor inquiries welcome

STYLEWriter

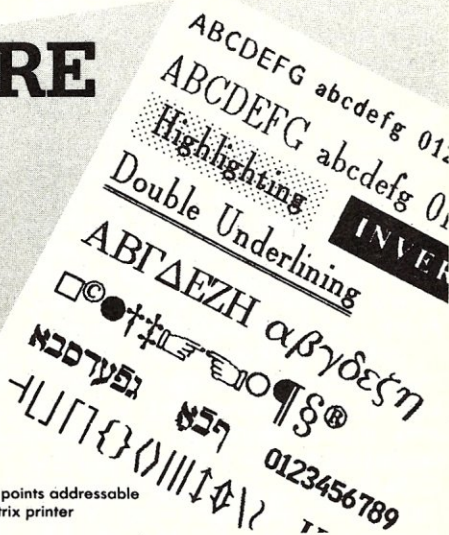
For only \$59 extra ...
this high
quality buffer
gives you so much MORE



- Upgrade dot matrix printers* to letter quality
- Over 100 types styles
- Double underlining, highlighting, inverse printing and right justification
- Foreign alphabets, scientific and mathematical characters
- Works with any computer, any software
- Basic stylewriter buffers come in 8K (\$139), 64K (\$169) & 128K (\$199) versions



For more information call 1-800-222-9073, (704) 525-4423
or write • 818 Tyvola Rd./109 • Charlotte, NC 28210



*Any all points addressable
dot matrix printer

Build A Smart Keyboard Interface

by John Monahan

Ever wished your keyboard had more keys or all your programs used the same control codes for cursor movements and editing functions?

This has always been my problem! Recently I decided to do something about it. After all, if I have my own personal computer, I should be able to have it do the things I want to do. I happen to have an S-100 system with Z80 and 8086 CPU's, and since I use an IBM-PC compatible keyboard and run MS-DOS, I wanted to run MS-DOS software that looks for an IBM keyboard.

My approach, which can be added to many other types of systems, is to have the keyboard talk to a keyboard controller which in turn talks to the computer. By having intelligence between a dumb keyboard and the computer, a truly powerful system can be set up, because the controller does not take up CPU space; it is always present, even on power up, works with different operating systems, and, as we shall see, can be much more powerful than its software counterpart.

THE CIRCUIT

To have any kind of flexibility, one must put together a unit containing an 8-bit microprocessor, a few I/O ports, as well as RAM and ROM memory. There are a number of very good single-chip, 8-bit microprocessors that have these features. The Intel 8048 is a classic example. However, in order to put something together quickly in hardware and to program it with an assembler I already had, I used the popular Z80. It requires few parts to assemble a very powerful system. A Z80 CPU with a 2716 ROM, 2Kx8 RAM, and two Zilog PIO's, together with a few small "glue" chips are all we need to construct a computer with 4 parallel ports with handshaking. Figure 1 shows the complete computer.

Because Z80 output pins are TTL compatible and capable of driving one TTL load, no buffers are required on the address, data or control lines. The 11 lines needed to address the 2716 ROM and 2Kx8 RAM chips can be connected directly to the Z80. These correspond to address lines A0 to A10. Address line A11 selects the

Add Intelligence To Your Keyboard Input

memory in the ROM (A11 = 0) or RAM (A11 = 1). Pin 20 is the critical pin for selecting the 2716 ROM. Whenever this (CS*) pin is low, the ROM places data on output pins: 9, 10, 11, 13, 14, 15, 16, and 17. These are then read by the data-bus pins of the Z80 when the Z80 brings MEMR*, RD*, and A11 low. Any write-to-RAM-memory or R/W-to-an-I/O port will not cause the 2716 CS* to go low and so will not cause it to place data on the data bus.

The operation of the 2K RAM chip is a bit more complicated. We have two possibilities. First a read from the memory; Pin 21 (WE*) should be high. No problem here since it is connected to the Z80 WR* line. Pin 20 (OE*) of the RAM chip is connected to the Z80 RD* line. When this is low, along with the Z80 MREQ* and a high Z80 A11 line, the 2K RAM places the selected data (determined by address lines A0 - A10) on the data bus to be read by the Z80. For a write to RAM, memory-pin 20 (OE*) of the RAM must be held high. Pin 21 (WE*) is brought low by the Z80 WR* line. The CE* is selected as described above for the read cycle.

Because we are using only 8, of the possible 256 I/O ports available on the Z80, we do not have to decode the address lines completely to address the two Zilog PIO's. The PIO's are Zilogs' answer to the Intel 8255. They are 40-pin LSI chips that contain 2 separate 8-bit parallel ports with handshaking. They may be programmed in a number of configurations. More on this later.

The PIO's have a unique built-in capability to interrupt the Z80 in an ordered manner. We will not be using this characteristic of the PIO, however this

requires that the Z80 clock and M1 pins to be directly connected (see Figure 1).

Each PIO is selected by bringing pin 4 (CS*) low. We do this on PIO1 and PIO2 by lowering address lines A2 and A3, respectively. Because each PIO contains 4 ports, we use address lines A0 and A1 to select these. Pin 5 of the PIO selects the command or data port. Pin 6 selects either port A or B on the chip. Address lines A0 to A3 will always be changing as the CPU reads memory, however, only when the IORQ* line goes low, will the PIO be addressed. This only happens when the Z80 code forces the CPU to read or write to a port.

The Z80 requires an external clock signal. Anything from 2 to 4Mhz is fine for this application. This is provided by the simple oscillator circuit connected to pin 6 of the Z80. Note that a 600Ω pullup resistor is required, since Zilog specifies the voltage swing must be within 0 to 5 volts.

CHECK OUT

So far we have assembled the bare essentials of a computer. With the appropriate software in ROM, the Z80, after power up, can be made to look at one data port and transfer the data across to another port. Complicated character translations can be done by adding on software as described below. If this is the first time you have put together a system of this complexity, you may first want to try something simpler first: namely fill a 2716 ROM with 76h's (the Z80 HALT instruction), switch on the power, and check that pin 18 of the Z80 has gone low, indicating the CPU has gone into the halt state. If this does not happen, more than likely you have one of your address or data lines connected incorrectly.

INTERFACING TO THE KEYBOARD

Connecting the Z80 board to an IBM-like keyboard entails one complication. The keyboard sends the data serially over two wires. One contains the data as 8-bits, the second, the keyboard clock data associated with the data. It would have been nice if IBM had chosen to use a standard UART-compatible, serial-data

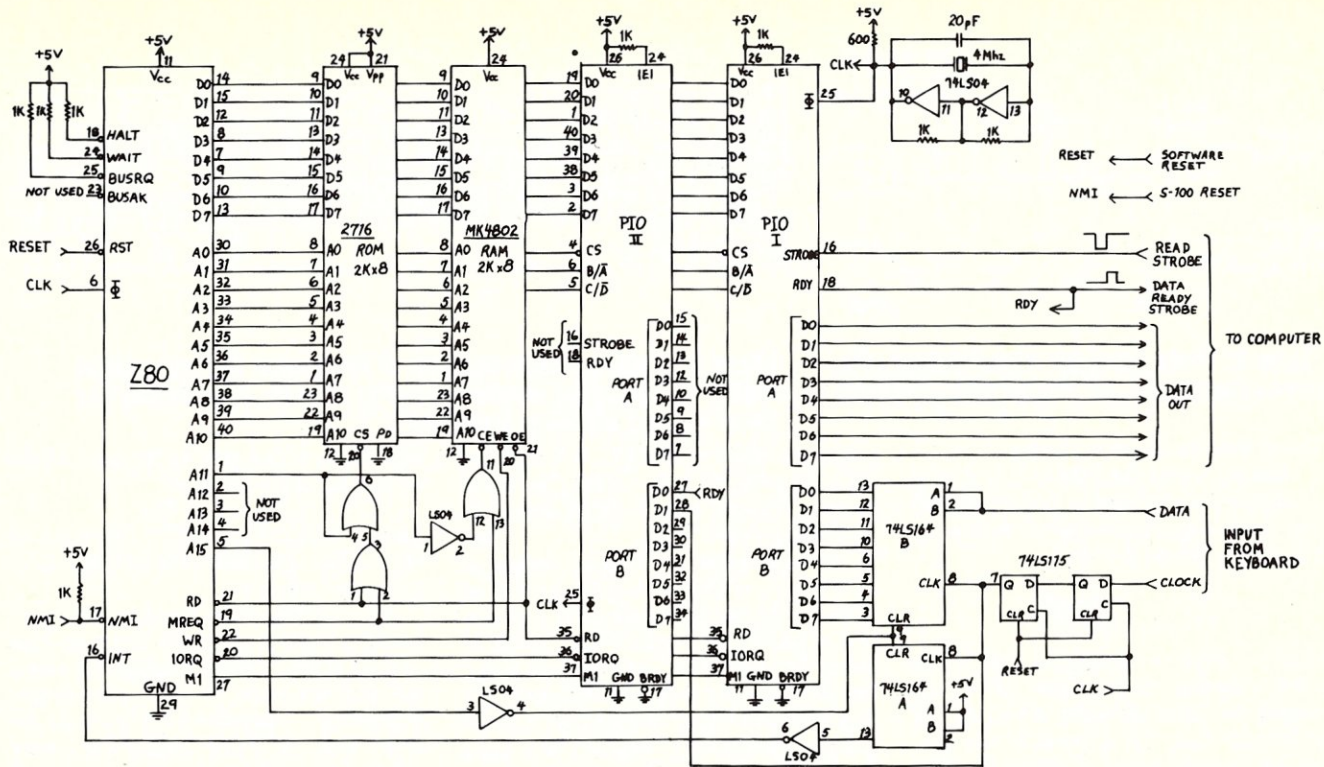


Figure 1. Keyboard controller circuitry

format. No such luck. The data is sent as 8-bits with no start bits or stop bits (Figure 2).

While it would be possible to program the Z80 to monitor one bit of an I/O port to assemble a byte from the serial data, the hardware solution of using two 8-bit shift registers makes life so simple that I opted for the easy way out. Here is how it works. Eight serial clock bits are shifted into the LS164-A-register (Figure 1) from the raw clock line coming from the keyboard. At the same time, 8 data bits are shifted into position in the LS164-B-register. Pins 3, 4, 5, 6, 10, 11, 12, and 13 of this chip will end up with the data in parallel form. The rising edge of the final clock bit raises pin 13 of the LS164-A-register. This, via the LS04 inverter, causes the INT* input to the Z80 to go low. This in turn causes the Z80 to call on an interrupt in ROM, which will pulse address line A15 high, clearing the two LS164 shift registers and readying them for the next byte from the keyboard. The raising of A15 is a cheap way of getting a fast 1-bit output port. Since our computer has no memory above 4K, we do not have to worry about the value of the high-order address lines. Writing a byte to address F000h in RAM will raise and lower A15 with no damage to RAM contents in low memory. The LS175 dual flip-flops (IC A) aligns the clock information with the data from the keyboard.

We are almost there, hardware-wise! All that remains to be done is to have the Z80 process the data and pass it on to the main computer. On my system, the key-

board input is from a SD Systems 8024 Video board. This board requires a keyboard with a parallel port. The data must be strobed into this port by a positive-going pulse. I have modified this board slightly so that once data is read from this port, a negative-going pulse is sent back to the smart keyboard interface letting it know the data has been read. Other boards may have different strobe protocols. These can often simply be accommodated by one or two 74LS04 inverters in the circuit. For those computer systems that have a serial keyboard, you could replace one of the PIO's with a Zilog SIO. This is a simple hardware replacement, but you should carefully study the software setup procedure to talk to the SIO.

Pin 26 of the Z80 (reset) is connected to the main-computer reset line. In this way, a reset to the main computer also resets the smart keyboard circuit. Any keyboard characters in a queue or taken through a translation table are flushed out in this process. The NMI* pin of the Z80 is connected to a one-bit output port of the main computer. The function of this will become clear when we discuss the software used to drive this board. Other pins of the Z80 are left either unconnected or tied high via a 1K resistor.

SOFTWARE

Writing software for a computer like this is a lot of fun. Because you have complete control of the smart-keyboard interface Z80 at all times, you can place values in certain registers and know they are always going to be there. We can, for ex-

ample, really make use of the Z80 alternate registers. The software I used to program this board has been submitted to the public-domain SIG/M users group and will be available in one of their future releases. What I would like to do, is step you through the main points. The complete details for all routines can be found in the public-domain code itself. The first two lines of code start off:

```
ORG 00H
LD SP,STACK
```

Since the Z80 reads an opcode from memory location 0, at power on or a reset, the ROM code must originate here. First we need a valid location for the stack. To be on the safe side, we will put the stack high up in RAM, but just below certain reserved RAM memory locations. I have used STACK = 0FF0h.

Next we initialize the Zilog PIO's. The 2 PIO's have 4 ports which can be configured in software as input, output, or bidirectional. Consult the Zilog technical literature for a detailed explanation of what this entails. We will set up both PIO port A's as output ports (MODE 0) and port B's as input ports (MODE1).

The PIO's also have the capability of generating an interrupt under certain data-transfer conditions. We do not need this here and so we must program the chips to disable this function.

Programming the chip is easy. You select the appropriate PIO control port (data port + 1 in this example) and send two bytes of code. Since we have 4 ports in all, we must send 2 bytes to each of 4 ports.

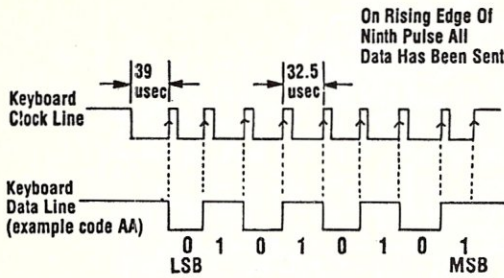


Figure 2. Timing diagram of serial data sent from a Keytronics IBM-compatible keyboard.

This is done as shown in Listing 1.

Having set-up the PIO's, it is a good practice to clear them of any false data they may have acquired before or during initialization. We do this by:

```
IN      a, (DATA$B)
IN      a, (DATA$C)
```

Next we have to enable the Z80 in the correct interrupt mode. Again you should consult the Zilog literature if you do not understand how this is done. In our case we need interrupt mode 1. This will cause the CPU to jump to location 38H in our ROM anytime the INT* (pin 16) on the Z80 is pulled low. At that location will be the code that will get data from the PIO port and process it. An INT will occur only when the hardware has received 8 serial bits of data from the keyboard (the low at pins 1 & 2 of the 74LS174-B have been shifted 8 times). The code at 38H is shown in Listing 2.

What we are doing here, is quickly taking the data byte at the keyboard and placing it in a cyclic 256-byte buffer in RAM memory. We do not know when such an INT would occur and so cannot count on what is in the "main" Z80 registers. For this purpose, we set aside the alternative Z80 registers. As described below, the keyboard data port will ALWAYS be in register C' and HL' will ALWAYS point to the end of the incoming character queue. One nice thing about the INC L, is that it insures the queue will always wrap around after 256 bytes. We do not have to move pointers back to the beginning of the queue once they reach its end. We have already discussed the use of address line A15 to reset the 74LS164's. We waste the IY reg to do just this in this application. It is setup with the value F000h. As you can see from the timing diagram of the clock and data lines in Figure 2, there is one extra clock pulse we have to absorb, before the keyboard is finished sending its byte of data. This is monitored at bit 1 of the PIO 2, port-B bit 1 (Figure 3).

At org 100H in the ROM, we have the code (Listing 3) to set-up the computer and keep it happy while it is waiting for a character.

What we have done in the above code is

3B BB	3C BC	01 81	02 82	03 83	04 84	05 85	06 86	07 87	08 88	09 89	0A 8A	0B 8B	0C 8C	0D 8D	0E 8E	45 C5	46 C6
3D BD	3E BE	0F 8F	10 90	11 91	12 92	13 93	14 94	15 95	16 96	17 97	18 98	19 99	1A 9A	1B 9B	1C 9C	47 C7	48 C8
3F BF	40 C0	1D 9D	1E 9E	1F 9F	20 A0	21 A1	22 A2	23 A3	24 A4	25 A5	26 A6	27 A7	28 A8	29 A9	2A AA	49 C9	4A CA
41 C1	42 C2	2B AB	2C AC	2D AD	2E AE	2F AF	30 B0	31 B1	32 B2	33 B3	34 B4	35 B5	36 B6	37 B7	38 B8	4B CB	4C CC
43 C3	44 C4	39 B9														4D CD	4E CE
																4F CF	50 D0
																51 D1	52 D2
																53 D3	54 D4
																55 D5	56 D6
																57 D7	58 D8
																59 D9	5A DA
																5B DB	5C DC
																5D DD	5E DE
																5F DF	60 E0
																61 E1	62 E2
																63 E3	64 E4
																65 E5	66 E6
																67 E7	68 E8
																69 E9	6A EA
																6B EB	6C EC
																6D ED	6E EE
																6F EF	70 F0
																71 F1	72 F2
																73 F3	74 F4
																75 F5	76 F6
																77 F7	78 F8
																79 F9	7A FA
																7B FB	7C FC
																7D FD	7E FE
																7F FF	80 00

Figure 3. Keyboard code chart for a Keytronics IBM-compatible keyboard. The upper number on each key is for downstrokes and lower number is for key upstrokes.

set-up the register pairs HL, DE, BC, and HL' to point to two regions in RAM memory that will contain the incoming and outgoing keyboard data.

Here is what will happen. When an INT occurs, a keyboard character will be placed at the end of a queue in the inbuffer. The pointer to this queue (in HL') will be increased by one byte and a 0 placed in that memory location. When the Z80 is not getting more characters from the keyboard and placing them in the inbuffer or sending them to the main computer by reading from the outbuffer (see below), it is checking if its pointer to the character in the inbuffer is zero. If it is not, it assumes one or more new characters have arrived. These characters are read from the inbuffer, translated if need be (see below), and placed in the outbuffer. The pointers to both buffers are updated accordingly and a zero flag in each buffer is set to indicate the ends of the buffers. It is important to remember that this is all this Z80 will ever have to do. So, certain registers can be set aside permanently to hold certain values. The code is shown in Listing 4.

The subroutine TRANS is the heart of the code. It takes the bit pattern from the keyboard and translates it into ASCII characters. This is necessary because the IBM/Keytronics keyboards sends only a binary number representation of the key pressed, not the ASCII character. For example, the ESC key sends 01H, the "1" key 02H, etc. Further, the keyboard distinguishes between key down-strokes and key up-strokes. Up-strokes have the same binary number plus 80H. In other words, their most significant bit is set. This is not all as bad as it sounds. It means we must make a lookup table of ASCII characters from binary values. Figure 3 shows the way the keys are numbered on an IBM/Keytronics keyboard. Part of the corresponding table looks like:

```
IBMTBL: DB      0
          DB 1BH, '1234567890-=', 8H
          DB 9H, 'qwertyuiop[]', 0DH
          DB 42H, 'asdfghjkl;', 27H, 60H
          DB 41H, 'zxcvbnm,./', 41H, '*'
          DB 1EH, ' ', 44H
```

When TRANS arrives, with say a 02H in register A, the actual ASCII value is obtained by adding 2 to the [HL] register

pair which is pointing to the start of the table. Then an instruction:

```
LD      A, (HL)
```

places the correct ASCII character in register A. The code for TRANS is as shown in Listing 5.

For the Smart Keyboard to be of use, we need a number of tables of the type described above. This is because the meaning of a key-board character can change, depending on whether keys such as the shift, lock, NUM or control keys were previously pressed. Each time one of these keys is pressed the appropriate flag is updated in RAM to set our [HL] pointer to the appropriate table. Rather than present all this code, I have sent it to the public domain SIG/M library. The file SKEY.Z80 contains all the code described in this article.

Besides the above special keys, which almost every keyboard has, we can add new ones. For example, we might have a case in which the F1 key is pressed; we have TRANS point to a table which defines the keys of the number keypad as special control sequences for a word processor such as Wordstar. F2 would point to a different table for a text editor such as Vedit. For complete compatibility with the IBM-PC, we can have a table where untranslated information (binary key numbers) is sent to the BIOS of the computer and translated exactly as IBM describes.

Now for the most important feature of the board — single-key to multi-key translation. If TRANS observes that the translated key is greater in value than 7FH (bit 7 high), another routine which I have named MULTI, is called. This routine looks at the "special character" and depending on its value, places not one, but a string of characters in the outbuffer. This string is then read by the main computer which thinks they were individually typed. To give you an example: If I press the "F6 key" TEST.TXT on my keyboard - 9 key strokes + CR, the computer would receive VEDIT TEST.TXT. At the same time, the numeric keypad would be automatically configured for the Vedit cursor/editing control sequences. This is done by pointing TRANS to the appropriate lookup table.

Because we have the power of a micro-processor at our disposal, we can do a lot with one keystroke. For example, I like to have the same cursor control keys for all my editors and word processors. Vedit uses one control character in many situations in which Wordstar uses two. In fact, in some cases, one needs to toggle two sets of dual control characters in Wordstar to get the same effect as with Vedit. This can be easily accommodated with this setup. The routine MULTI is quite long and contains a number of special-case treatments for special keys. There is not room here to present the whole routine. However the kernel of the routine is shown below in Listing 6.

In this routine, two data areas are used. Multi\$table contains a list of pointers to the first character of each string. The offset into multi\$table X2 will allow the correct pointer to be picked up. This in turn points to the actual string which can be of varied length. A simplified version of the table is shown in Listing 7.

The DROP routine then transfers each character of the string into the outbuffer until a 0 is reached.

One final point. When you are finished with your special application program, it would be nice to reset the keyboard back to its default configuration (CP/M or MS-DOS). It would be nice also not to have to do this manually. This is where I have utilized the NMI line to the Z80. Whenever this line is pulled low, the Z80 stops whatever it is doing and jumps to 66H in RAM where it finds the code shown in Listing 8.

Now, while this is not entirely clear unless you read the complete code, suffice to say, all flags set up in memory are reset to their initial power-on (CP/M or MS-DOS) configuration. How does the NMI line get triggered? This is where you have to use your own initiative. In my S-100 system, I use one bit of an output port to lower and then raise the NMI line on the Z80 board. In the BIOS portion of CP/M+ and CP/M86, one can put the required code in the warm-start module just before control is transferred to the CCP.

For MS-DOS, I utilize the fact that the command-line prompt is definable. For example it could be "A>" or "A\$". I set it to "Esc Esc A>". When my console output driver sees the unique "Esc Esc" sequence, it sends a signal to the smart keyboard to pulse the NMI line of the Z80. There are many other possibilities. Many editors have logoff control sequences, indeed, you could set aside one keyboard key for the function.

Lastly, the power of the Z80 in this application is hardly used. You can easily modify the design to use keys to switch-on drive motors, CRT's, or to restrict access to certain programs. I have mine connected to a speech synthesizer that sends me all kinds of information and reminders. [E]

Save

BUILD YOUR OWN IBM XT & IBM AT COMPATIBLE SYSTEMS

Save

-Why Pay More-Build Your Own With Ease-Have Fun-Save a Fortune-
Introducing Super XT-16 Self-Assembly Kit
Assembled in Less Than 1 Hour with Screw Driver at SUPER LOW COST

- Including 640K XT-16 CPU Mother Board, 256 Installed, Color Graphic Card or TTL Monochrome Card, Floppy Disk Controller Card, One ½-height DS/DD Drive, Flip-Top Case, 135W Power Supply, Keyboard, Assembly Instruction and User's System Manual.

ONLY
\$695.00

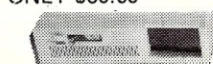


- We will assemble at no charge.
- Turbo Kit — Add \$50

XT, AT CASE

- Same Dimension as IBM PC/AT
- For IBM PC/AT Compatible Mother Boards
- Flip-Top For Easy & Quick Access to Inside

\$119



XT, AT MOTHER BOARD

- XT-16 CPU Mother Board - \$167.00
- IBM PC/XT Fully Compatible, Run all Popular IBM Softwares
- 8088 Microprocessor w/8087 Optional
- 8 I/O Slots
- Up to 640K Memory on Mother Board
- Fully Assembled & Tested
- AT-32 CPU MOTHER BOARD
- IBM PC/AT Fully Compatible
- 80286 Microprocessor w/80287 Optional
- 640K Standard, Upgrade to 1 MB on Board
- On Board Clock Calendar
- 8 I/O Slots

\$795

XT, AT POWER SUPPLY

XT-135W \$105.00 AT-200W \$169.00

XT, AT KEYBOARD

XT-LED for Cap Lock & Num. Lock
84 Keys \$70



AT-Same Layout as IBM PC/AT \$109

TURBO XT MOTHER/BOARD

- TURBO XT 640K MOTHER/BOARD w/8 Slots IBM compatible. Runs at XT & AT speed. Fully TESTED & ASSEMBLED. W/BIOS - OK \$195.00

PC/XT ADD-ON CARD

- Disk IO Card \$139
- Turbo XT M/B \$195
- 384 Multifunction Card - Serial, Parallel, Clock, Game, w/Cables & Software - OK \$109
- Mono/Graphic w/Printer Card - Version II Hi-Res (Lotus 1 2 3) . \$109
- Color Graphic Card w/Manual . \$87
- Monochrome TTL Card \$70
- 512K RAM Card OK w/Manual . \$65
- Floppy Disk Controller w/Cable (handle 4 drives) \$55
- RS232C Card w/Manual \$40
- Parallel Printer Card \$30
- Game Card (supports 2 Joy Sticks) \$25

All Cards Fully Tested, Assembled & Warranted / School & Institutional R.O. Accepted
OEM Dealers Welcome - Please call for our Special Dealer Prices

ATLAZ COMPUTER SUPPLY

616 Burnside Ave. / Inwood, NY 11696 • Mail Order Hotline 516-239-1854

(IBM is a trademark of International Business Machines Corporation)

LISTING 1

```

OUTBLOCK: LD HL,PIOTBL ;POINT TO THE PIO TABLE
OR A, (HL) ;GET BYTE COUNTER
JR Z,ALLSET ;TEST FOR END OF TABLE
LD B,A ;0 WHEN ALL PORTS DONE
INC HL ;ELSE PUT COUNT IN B
LD C, (HL) ;POINT TO PORT #
INC HL ;PUT PORT IN [C]
OTIR ;POINT TO FIRST DATA VALUE
JR OUTBLOCK ;SEND 2 BYTES TO PORT IN [C]
;GO TO NEXT PORT.

```

Continued on next page.

John Monahan is a molecular biologist. He received a PhD in Bio-Chemistry from MacMaster University in Ontario Canada. John has been a computer hobbyist for the past ten years and has built several homebrew

systems. He has been a member of the Amateur Computer Group of New Jersey for over 9 years and recently moved to the San Francisco bay area. He can be contacted at: Box 1908, Orinda CA 94563.


```

EXX      ;SETUP THE REGS FOR INT PROCESSING
LD        HL,INBUFFER ;[HL'] POINTS TO BUFFER
LD        C,DATASB    ;DATA PORT FOR KEYBOARD
LD        (IY+0),A    ;CLEAR THE 74LS164'S
EXX      ;CAN NOW ENABLE INTS.
EI

```

```

PIOTBL:- DB 2 ;Number of bytes to send
          DB PORT$A ;value of port A
          DB 0FH ;mode 0
          DB 03H ;No Interrupts.
          DB 2 ;Number of bytes to send
          DB PORT$B ;value of port B
          DB 4FH ;mode 1
          DB 03H ;No Interrupts.
          DB 2 ;Number of bytes to send
          DB PORT$C ;value of port C
          DB 0FH ;mode 0
          DB 03H ;No Interrupts.
          DB 2 ;Number of bytes to send
          DB PORT$A ;value of port D
          DB 0FH ;mode 0
          DB 03H ;No Interrupts.
          DB 0 ;end of table flag

```

LISTING 2

```

;SAVE ALT REGS
CLOCK: EXX      AF,AF'
EX      A,(C) ;DEPOSIT CHARACTER AT END OF QUEUE
IN      (HL),A
LD      L ;POINT TO NEXT BYTE OF 256 BYTE QUEUE
INC     A ;FLAG END OF BUFFER
XOR     A
LD      (HL),A
CLOCK1: IN      A,(DATA$D) ;WAIT FOR LAST CLOCK PULSE TO RISE
BIT     1,A
JR      NZ,CLOCK1
CLOCK2: IN      A,(DATA$D)
        BIT     1,A
        JR      Z,CLOCK2
        LD      (IY+0),A ;RAISE A15 LINE TO RESET 74LS164'S
        EX      AF,AF' ;RESTORE ALT REGS
        EXX     ;ENABLE ANY MORE INTS
        EI      ;RETURN FROM INTS
        RETI

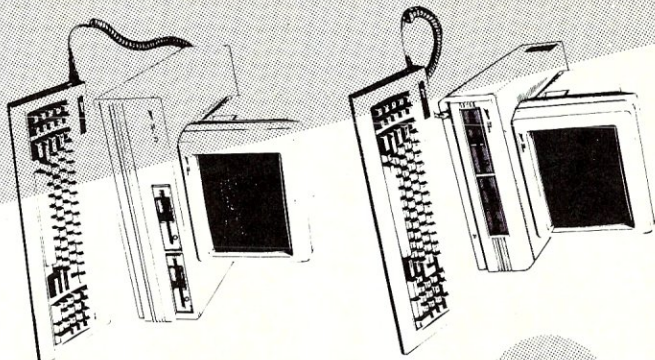
```

LISTING 3

```

BEGIN: XOR     A ;FLAG TO INDICATE KEYS IN SHIFT MODE
LD      (SHIFTFLAG),A
LD      IY,0F000H ;FOR A15 TO CLEAR 74LS164'S
BEGNMI: LD      HL,INBUFFER ;BUFFER FOR INCOMING CHARACTERS
LD      (HL),A ;ZERO AT END OF BUFFER
LD      (WORD1),A ;SPECIAL KEY FLAGS USED BELOW
LD      (WORD2),A
LD      (FLAG1),A
LD      (FLAG2),A
LD      DE,OUTBUFFER ;END OF OUTBUFFER QUEUE
LD      BC,OUTBUFFER ;START OF OUTBUFFER QUEUE
LD      (DE),A ;ZERO IT ALSO

```



Call *First Capitol* for the best prices and custom configurations on computers

Z-148 13 Meg hard disk system

- * Only \$1629 complete!
- * 640K, Monochrome monitor
- * 8 Mhz, more!

Call or write for new 40 page catalog!

Z-200 "SuperServer"—ideal as LAN fileserver or super CAD station.

- * 62 mb (formatted) 28 ms access drive
- * Network hardware included
- * High resolution video boards/monitors available
- * 69 Mb high speed tape backup
- * Backup power supply

CALL WITH YOUR REQUIREMENTS!

ZENITH data systems

AUTHORIZED SALES AND SERVICE

First Capitol Computer

1106 First Capitol Dr.
St. Charles, MO 63301
(314) 724-2336

From \$1049

ZF-148-41

LISTING 4

```

LOOP:  LD    A,(HL)      ;SEE IF ANYTHING IN INBUFFER
      OR     A
      JR     Z,LOOP1
      INC    L           ;IF NOTHING CHECK OUTBUFFER
      PUSH   HL          ;INCREASE POINTER FOR NEXT TIME
      PUSH   BC          ;SAVE THESE REGISTERS
      CALL   TRANS       ;CONVERT BIT PATTERN TO ASCII CHARS
      POP    BC          ;RESTORE REGISTERS
      POP    HL
      OR     A           ;DO NOT SEND NULLS TO MAIN COMPUTER
      JR     Z,LOOP1
      BIT    7,A         ;FLAG TO INDICATE SPECIAL CASES
      JR     Z,NORMAL    ;IF Z THEN SIMPLE ONE TO ONE
      PUSH   HL          ;NEED TO SAVE IT AGAIN
      CALL   MULTI       ;ONE IN CHAR TO SEVERAL OUT CHARS
      POP    HL
      JR     LOOPM

;
NORMAL: LD    (DE),A     ;PUT ASCII CHAR IN OUTBUFFER
      INC    E           ;FLAG END WITH ZERO
LOOPM:  XOR    A
      LD     (DE),A
;
;
LOOP1:  LD     A,(BC)    ;ANYTHING TO SEND TO
      OR     A
;
      JR     Z,LOOP      ;NOTHING THERE BACK TO INBUFFER QUEUE
;
LOOP2:  IN     A,(DATA$D) ;IS COMPUTER READY FOR NEXT CHAR.
      BIT    0,A
      JR     NZ,LOOP2    ;HANG IN THERE UNTIL READY
      LD     A,(BC)      ;[BC] POINTS TO CHAR
      INC    C           ;FOR NEXT TIME
      OUT    (DATA$A),A  ;SEND IT
      JR     LOOP

```

LISTING 5

```

;
TRANS:  LD     HL,SHIFTFLAG ;POINT TO FLAG OF CURRENT SHIFT ETC. MODES
      BIT    7,A           ;IS KEY AN UPSTROKE MOVE
      JR     NZ,UPSTROKE
      CP     SHIFT1       ;IS IT A SHIFT KEY
      JR     Z,ISSHIFT
      CP     SHIFT2       ;TWO SHIFT KEYS ON BOARD
      JR     Z,ISSHIFT
      CP     SHIFT3       ;FOR SHIFT LOCK KEY
      JP     Z,ISLOCK
      CP     NUMLOCK       ;IS IT NUMBER LOCK KEY
      JP     Z,ISNUM
      CP     CTRL          ;IS IT THE CTRL KEY
      JP     Z,ISCTRL
      PUSH   AF           ;SAVE CHARACTER FOR THE MOMENT
;
      LD     A,(HL)       ;FIND OUT WHICH TABLE IS CURRENT
      BIT    6,A          ;BIT 6=1 FOR WORDSTAR TABLE
      JR     Z,NOTSTAR
      LD     HL,STARTABLE
      JR     ENDSHIFT
NOTSTAR: BIT    4,A       ;BIT 4=1 FOR VEDIT TABLE
      JR     Z,NOTVED
      LD     HL,VEDTABLE
      JR     ENDSHIFT
NOTVED:  BIT    5,A       ;BIT 5=1 FOR IBM/DOS KEYBOARD
      JR     Z,NOTIBM
      LD     HL,IBMTABLE
      JR     ENDSHIFT
NOTIBM:  LD     HL,CPMTABLE ;X000XXXX = DEFAULT TABLE TO CP/M TABLE

```

```

;
ENDSHIF:AND 00000010B ;IS UPPER OR LOWER SECTION OF TABLE REQ
      JR     Z,UPPERHALF ;BIT 1=1 FOR UPPER CASE CHARS (!@#$...)
      LD     B,0
      LD     C,HALF
      ADD    HL,BC       ;HL NOW POINTS TO SECOND HALF OF EACH TABLE
UPPERHALF:
      LD     B,0
      POP    AF          ;GET CHARACTER BIT PATTERN
      LD     C,A
      ADD    HL,BC       ;GET OFFSET INTO TABLE
      LD     C,(HL)
      LD     HL,SHIFTFLAG ;STORE CHARACTER BACK IN [C]
      LD     A,(HL)      ;NOW NEED TO SEE IF WE NEED SHIFT MODE
      AND    00000110B   ;SEE IF a....z TO UC IS REQ
      JR     Z,ENDSHF    ;BITS 1 OR 2 =1 FOR UPPER CASE
      LD     A,C
      CP     'a'
      JR     C,ENDSHF
      CP     'z'+1
      JR     NC,ENDSHF
      SUB    20H
      LD     C,A
      ENDSHF: BIT    3,(HL) ;ARE CONTROL CHARACTERS REQ.
      LD     A,C          ;BIT 3=1 FOR CTRL CHARS.
      RET    Z
      AND    00011111B
      RET
;
UPSTROKE:CP UPSH1       ;WERE ANY SPECIAL KEYS RELEASED
      JR     Z,NOTSHIFT
      CP     UPSTROKE
      JR     Z,NOTSHIFT
      CP     UPCTRL
      JR     Z,NOTCTRL
      XOR    A           ;NONE THEN NORMAL RELEASE
      RET
;
ISSHIFT:SET 1,(HL)      ;IS SHIFT KEY SET FLAG
      XOR    A
      RET
;
ISLOCK:  XOR    A       ;TOGGLE CAPS LOCK ON/OFF
      BIT    2,(HL)
      JR     Z,TOGGLE
      RES    2,(HL)
      RET
TOGGLE:  SET    2,(HL)
      RET
;
ISNUM:   BIT    0,(HL)   ;THIS HAS TOGGLE INFO FOR NUMLOCK LED/BIT
      JR     NZ,TOGGL
      LD     A,(HL)
      AND    00001110B   ;IF 0, FORCE IBM TABLE
      OR     00100001B   ;SO NEXT TIME CP/M TABLE
      LD     (HL),A
      call   cleanflags  ;CLEAR VEDIT & WS FLAGS
      LD     HL,SPIEM
      CALL   TALK
      XOR    A
      RET
TOGGL:   LD     A,(HL)   ;IF 1, FORCE CP/M TABLE
      AND    00001110B   ;NOTE BIT 0 ALSO SET TO Z
      LD     (HL),A
      call   cleanflags
      LD     HL,SPCPM
      CALL   TALK
      XOR    A
      RET
;

```


Computer Language Magazine

R.D. Ashworth, Ph.D.

P.K. Wayne, MD, Ph.D.

Equipment: IBM-PC, AT, XT and most compatibles including those by Tandy, Compaq, Texas Instruments, Sperry, AT&T, Kaypro and others. **Memory:** Single User: 128K; Multi-User: 256K and up

"Cookbook of MUMPS" 180 page manual with accompanying disk. Includes dozens of fully documented routines and utilities, with examples of controlling output and display, global design hints, and mathematical functions. *RECOMMENDED*

Multi-tasking, Tool Run multiple concurrent background processes for data searches, report generation, etc.

call now for faster service

In Texas 713-529-2576

Please send me the following...

Single User disk and operations manual	\$59.95
"Cookbook of MUMPS" (includes disk)	\$24.95
SPECIAL Single user & "Cookbook"	\$75.90
Multi-tasking	\$149.95
Programmer's Toolkit I	\$49.95
Graphics option	\$49.95
Multi-user	\$450.00
MacMUMPS (Macintosh version)	\$89.95
shipping and handling	\$3.00
Texas residents add 6 1/8% sales tax	

card no. _____

name _____

street _____

city _____ st _____ zip _____

Disks are non-copy-protected

day phone _____

LISTING 6

LISTING 7

TESTING 8

```

ORG
NNMILOC: LD
LD
AND
LD
LD
XOR
JP

SP,STACK
A,(SHIFTFLAG)
00001111B
(SHIFTFLAG),A
A
BEGNNI

;GET SHIFT/TABLE FLAGS
;RESET ALL FLAGS

```


COMPETITIVE EDGE

631 S. Main St. — Plymouth, MI 48170 — (313) 451-0665

Competitive Edge brings you Lomas Data Products Inc.

S-100 CIRCUIT BOARDS

Thunder 186 , 256K, 2 Serial, 1 Parallel, Floppy Disk Controller, Clock, All on One Board with Concurrent DOS					\$895
8MHX Lightning 286 CPU	\$756	NV Disk 512K, Memory Drive	\$371	512K Dram Megagram	\$448
10MHZ Lightning 286 CPU	821	Concurrent DOS Single User	280	Thunder + 186 512K 10MHZ	895
Control It All Floppy & HD Controller	487	MSDOS™ for 86, 186, or 286	200	Thunder + 186 1024K 10MHZ	999
Hazital 2 Serial, 2 Par, Clock	260	CPM22EM 8080 Emulator	55	Octaport 4 Serial	200
256K Dram Megaram	358	6MHZ Lightning 286 CPU	756	NV Disk 2048K Wow!!	746
1024K (1 Megabyte) Megaram	599	8 MHZ Lightning 8086	420	Concurrent DOS Multi-user	360
Ram 67 128K Static 100ns Chips	596	LDP Floppy Contr.	220	LDP-COM Modem Program	55
Octaport 8 Serial to 38.4K	320	Color Magic Color Graphics Board	476	PC DOS Drivers	35

COMPETITIVE EDGE INTEGRATED SYSTEMS

THUNDER 186, 4 SLOT PC STYLE CABINET, 2-5" FLOPPYS, CDOS 512K	\$1595
THUNDER 186, 4 SLOT, 20 MB HARD DISK, 1-5" FLOPPY, CDOS 512K	2595
THUNDER 186, 4 SLOT, 20 MB HD, 1-5" FLOPPY 4 USER 1024K	3195
THUNDER 186, 4 SLOT, 2-5" FLOPPY, COLOR MAGIC, KEYBOARD 512K	2250
8MHZ 286, 1-5" FLOPPY, 1024K, 20MB HD, 10 SERIAL, 15 SLOT, 7-8 USERS	4695
8MHZ 286, 1-5" FLOPPY, 1024K, 40MB HD, 10 SERIAL, 15 SLOT, 7-8 USERS	5295
10MHZ 286, 1-5" FLOPPY, 1024K, STATIC, 40 MB HD, 10 SERIAL, 7-8 USERS	7550
10MHZ 286, 1-5" FLOPPY, 512K STATIC, 40 MB HD, 2 SERIAL & 80287	6495
8MHZ 8086/8087, 1-5" FLOPPY, 512K DRAM, 20 MB HD, 2 SERIAL CDOS or MSDOS	3695
10MHZ 286 1-5" FLOPPY 1024K 120MB HD 10 SERIAL SLOT 7-8 USERS CDOS	8495

LOMAS STATIC RAM NEW 256K TO 1024K PRICES START AT \$412

FEATURING CompuPro® FROM VIASYN™

S-100 CompuPro® CIRCUIT BOARDS

Disk 1A™ Floppy Disk Controller	\$446	Network 100™ Network Board	\$375	Interfacer 4™ 3 Serial 1 Par	\$339
8 MHZ 286 CPU A&T 2 CYL.	677	80287 Option for 286 CPU	199	CPU Z™ 8MHZZ80	207
Ram 22™ 256K Static Ram A&T	446	Concurrent DOS™ 8-16™	638	MDRIVE-H 2048K, 2MB	825
Ram 23™ 64K Static Ram A&T	189	CP/M® 2.28" or 5"	189	Keytronics 5150 for PC Video	169
Interfacer 3™ 8 Serial Ports	449	Disk 3™ Hard Disk Controller	525	Passive Hub for Network 100	65
SPU Z™ 8MHZ Z80, 256K Multiuser	296	10MHZ 8088, 85/88 CPU A&T	269	HX-12 Hi-Res Color Monitor PC	459
MDRIVE-H™ 512K Memory Drive	375	Ram 23™ 128K Static Ram A&T	263	CP/M® 8-16™	268
PC Video Board for IBM® Compat	399	System Support 1™ A&T	338	SP186 512K Slave	495

HARD DISK SUB-SYSTEMS

40 MEGABYTE HARD DISK, CABINET, DISK 3, SUB-SYSTEM	\$1950
80 MEGABYTE HARD DISK, CABINET, DISK 3, SUB-SYSTEM	2450
ALLOY 17 MB TAPE BACKUP	1795
40 MEGABYTE HARD DISK, CABINET, DISK 3, 5" FLOPPY	2150
8" & 5" FLOPPYS, 40 MB HARD DISK SUB-SYSTEM	2650
2-40 MB HARD DISK DRIVES IN CABINET, DISK 3, SUB-SYSTEM	3095

SYSTEMS INTEGRATED BY COMPETITIVE EDGE WITH COMPUPRO® COMPONENTS

8MHZ 286 SYSTEM

\$5895

8MHZ 286 & 512K STATIC RAM
9 SERIAL PORTS, SS1 & I/O 3
ONE 5" FLOPPY & 40MB HARD DISK
DISK 1A & DISK 3 CONTROLLERS
CONCURRENT DOS 8-16 MULTI-USER
15 SLOT CABINET, 30 AMPS + 8
EM8080 SOFTWARE 8 BIT EMULATOR
OPTIONAL 8MHZ Z80H SLAVES
COMPONENTS ASSEMBLED & TESTED TOGETHER, ALL INTERNAL
CABLES SUPPLIED

8MHZ 286 SYSTEM

\$6295

AS ABOVE BUT 80 MB HD INSTEAD OF 40 MB HD

10MHZ 8088-85/88 SYSTEM

\$4495

85/88 CPU & 128K STATIC
4 SERIAL PORTS & 1 PAR
SYSTEM SUPPORT & I/O 4
ONE 5" FLOPPY & 40 MB
CP/M-8-16 8 & 16 BIT
15 SLOT CABINET
5" FLOPPY WILL READ IBM
SYSTEM UPGRADEABLE TO MULTI-USER & Z80 SLAVES ALL

ALL PRICES SUBJECT TO CHANGE — SHIPPING EXTRA

IBM is a registered trademark of International Business Machines, CP/M, Concurrent DOS are registered trademarks of Digital Research Inc. MSDOS is a trademark of Microsoft. All above circuit board names are either registered trademarks or Trademarks of Viasyn Corporation. CompuPro is a registered trademark of Viasyn Corporation.

NO EXTRA CHARGE FOR VISA OR MASTER CHARGE

COMPETITIVE EDGE

631 S. Main St. — Plymouth, MI 48170 — (313) 451-0665

HIGH PERFORMANCE PRODUCTS RUNNING UNDER

TURBODOS™

TELETEK ENTERPRISES S-100 MASTER/SLAVE BOARDS

Teletex
Systemaster
\$495

Systemaster II
\$795

8MHZ SBC 86
512K - \$963

5 MHZ SBC 86
512K - \$795

6MHZ SBC-1
128K - \$350

HDC
\$375

4MHZ Z80A MASTER SYSTEMASTER I, 2 SERIAL, 2 PAR, FOC, CLOCK	\$495
8MHZ Z80H 128K MASTER, SYSTEMASTER II®, 2 SER, 2 PAR, FDC CLK	795
8MHZ 8086 512K SLAVE SBC 86 RUNS WITH EITHER MASTER ABOVE	963
5MHZ 8086 512K SLAVE SBC 86 RUNS WITH EITHER MASTER ABOVE	795
8MHZ 8087 OPTION FOR 8MHZ 8086 SBC 86	175
5MHZ 8087 OPTION FOR 5MHZ 8086 SBC 86	109
SBC 1 6MHZ Z80B 128K SLAVE RUNS WITH EITHER MASTER ABOVE	350
HDC HARD DISK CONTROLLER — ST506 INTERFACE RUNS WITH ABOVE	375
SBC II DUAL SLAVE BOARD (TWO Z80'S PORTS ETC.)	689

COMPETITIVE EDGE INTEGRATED SYSTEMS

4MHZ Z80A 64K, CPM 2.2, 2-5" FLOPPYS, 4 SLOT CABINET	1295
8MHZ Z80H 128K, SINGLE USER TURBODOS WITH SPOOLING, 2-5" FLOPPYS, 4 SLOT CABINET READY TO RUN	1695
8MHZ Z80H, 128K MASTER, 4-8MHZ 128K SLAVES, 1-5" 20MB HD 10 SLOT 1-5" FLOPPY READY TO RUN WITH TURBODOS	\$4795
8MHZ Z80H, 128K MASTER, 4-8MHZ 512K SLAVES, 1-5" 20MB HD 10 SLOT 1-5" FLOPPY READY TO RUN WITH TURBODOS	\$7495
4MHZ Z80A, 64K MASTER, 4-6MHZ 128K Z80B SLAVES. 1-5" 20 MB HD, 10 SLOT 1-5" FLOPPY READY TO RUN WITH TURBODOS	\$4395

SYSTEMS AVAILABLE WITH UP TO 16 SLAVES AND 2-112 MD HD (224 MB TOTAL)

All Prices Subject to Change — Shipping Extra, Visa M/C Accepted

Systemaster I & II, HDC, SBC I, SBC II are either registered trademarks or trademarks of Teletex Enterprises. CPM is a registered trademark of Digital Research Inc., Turbodos is a trademark of Software 2000 Inc.

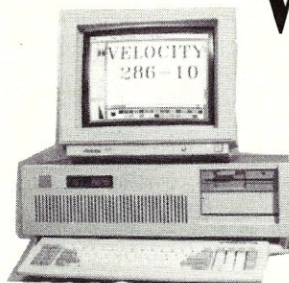
COMPETITIVE EDGE

631 S. Main St. — Plymouth, MI 48170 — (313) 451-0665

INTRODUCING

VELOCITY 286-10™

A T™ COMPATIBLE



10 MHz OPERATION

1024K on MOTHERBOARD

16 BIT HARD DISK TRANSFERS

SERIAL, PARALLEL, CLOCK

Enhanced Graphic Adapter Now Standard Runs with Color or Mono Monitor

1.2 MB Floppy, A T™ Compatible Keyboard, DOS 3.1

30 MEGABYTE FAST HARD DISK

PRICES START AT \$3295

DEALER & QUANTITY PRICING AVAILABLE

SHIPPING NOW!!!

— OPTIONS —

80287 Math CoProcessor

Enhanced Graphics Adapter

30 to 100 MB Hard Drives

20 to 100 MB Streaming Tape Back (2 minutes per 10 MB)

CPM®/Z80H Emulator Card Only **\$395.00**

8MHZ 8088 Slaves (up to 31) for Multi-User

3 MB Memory Expansion Only **\$795.00**

Monochrome & Color Monitors

Basic Xenix® 1024K, 8 Serial, 30MB System **\$4095**

NO EXTRA CHARGE FOR VISA OR MASTER CHARGE

CPM is a registered trademark of Digital Research Inc. Hercules is a trademark of Hercules. A T is a trademark of International Business Machines. Velocity 8 and are a trademark of Competitive Edge. Screen is PC Paint Brush copyright and product of 2Soft, Inc. Xenix is a trademark of Micro Soft.

The CP/M Bus

by Robert A. Blum

The CP/M Bus Bulletin Board System, operated by Bob Blum, is available for your use 7 days a week, 24 hours a day at 1200 baud. Reach it by calling (404) 449-6588.

Gaining enough experience to easily maneuver about the CP/M Bus Bulletin Board system, or any other BBS for that matter, can be a very time consuming project. Evidence to this fact is brought to my attention practically every day by way of questions that are left on my BBS system. And contrary to what you might think, many of the questions come from experienced users confused by the subtle differences that exist between BBS systems.

The root of this problem stems from the numerous ways that the same basic set of system control programs operating on most BBS systems can be configured by the System Operator (SYSOP) to suit their particular needs. For example, most SYSOPS want their system to be as impervious to abuse as possible while still providing the caller complete access to the treasures within. To this end, some SYSOPS demand complete control over system access by requiring users to pre-register by mail or phone before entry to the system is allowed. On the other hand, SYSOPS like myself don't want to deny system access to anybody that has at least a working knowledge of CP/M and can answer a question or two while online about its standard components.

As I've already said, either type of system can be setup using the same set of programs; but to a large degree, how the system responds to your commands will differ from system to system. Of course, this does not pertain to every system in operation. There are probably as many different programs running BBS systems as there are SYSOPS, but the majority of them will have at least some common commands and a comparable feel.

To get a basic understanding of what is needed to access a BBS and what some of their differences are before committing yourself to the phone company, one should read a reference book.

THE FREE SOFTWARE CATALOG AND DIRECTORY

Public Domain Software is software that can be legally copied and used and given to another without payment of a royalty to its author.

Estimates project that over 40 megabytes of program source code is contained on the 300+ volumes of CP/M public domain software. In this vast library can be found just about any type of utility program that you might need to help manage your CP/M system. And a number of business application programs, mostly written in Cbasic, are present as well.

Until now many would-be avid public domain software users have been stymied by the thought of contending with the over 5000 program titles in order to find just the few desired. Fortunately, Robert A. Froehlich, a long term member of SIG/M and avid user of public domain software, foresaw this problem and has compiled a reference book detailing and indexing each program.

The Free Software Catalog And Directory is a 475 page 8½ X 11" book printed on newsprint stock focusing on the two largest and most readily available free software libraries, the CP/M Users Group (CPMUG) and the Special Interest for Microcomputers (SIG/M), and provides all the information necessary for personal computer owners to take advantage of this resource.

The introduction begins by setting aside any fears one might have about using public domain software. It concludes by talking a little about the main sources of free software. But most important is the elaborate tutorial, the best I have seen, of successfully completing a session with a Remote Bulletin Board System (RBBS). This section in itself makes the book a worthwhile investment for the novice and experienced user alike. The remainder of the introduction prepares the reader for dealing with source code files and how to go about making changes to them should that be necessary.

The real meat of this book is contained in the next six sections. Every file in the entire 92 volume set of the CPMUG library and the first 162 volumes of the SIG/M library are described in detail. The first and second sections are the largest in size because complete information for each file is included as follows: file name, size in K bytes, CRC checksum, date of entry, language if program source code, authors name, revisors name, file title, keywords assigned to this file and finally a textual description of the file.

But, of course the main purpose of this book is to provide ready access to information about the routines or programs of

interest to you. The next four sections cross reference each file alphabetically by keyword, language type, author name, and lastly by file name. For example, if you wanted to find a program to compare two files for equality your initial search might start in the keyword section under utilities. Found there are two entries; the one for comparing binary files sounds like it would serve well; it is on SIG/M volume 115. Thus, a complete description of the chosen file can be found by referring to the second section of the book completely describing each SIG/M volume.

As if all this weren't enough, the book concludes by listing several hundred User Groups and bulletin board systems by name, location, and phone number.

This book has become the most used of any in my bookcase. It is a tremendous bargain at \$9.95. The author has accomplished a phenomenal task in compiling this much material and cross indexing it for ease of access. No matter whether you are an individual user of free software or the librarian of a user group you will benefit from having the book.

If you're having trouble finding a source of free software in your local area call the CP/M Exchange RCP/M system at (404) 449-6588. There you will be able to view a list of user groups where public domain software can be found.

MAKING CP/M BETTER

From Neil Harrison of London, England comes the following letter describing his endeavors at making CP/M and ZSID better programs:

I have been running the Plu*Perfect Systems Public patches for the CP/M 2.2 BDOS for a few months now and really find them a great help on my hard disk system. I have a suggestion for people using the Public file system which they may find useful.

One thing that has occasionally been a source of annoyance over the years is the inability of a Submit batch job to continue after changing the user number. The reason for this is fairly straight forward; when the user number gets changed the \$\$\$SUB file which contains further commands gets *left behind* in the previous user and the CCP fails to find it. Only when you return to the original user where the Submit job was started does the batch processing continue.

Press RETURN for more; type NO to stop:

The answer to this problem is for the

CCP to be able to find the \$\$\$SUB file from any user, i.e. for it to be a Public file. Now, since Public files are simply distinguished by having the F2 attribute bit set (bit 7 of the second character of the filename) all that is required is for SUBMIT.COM to be patched to create \$\$\$SUB with this bit set. The single byte patch to accomplish this is outlined in Listing 1.

The idea behind this patch should apply to all the superior Submit replacements that are in the public domain. Simply find the ASCII string "\$\$\$SUB" in memory using DDT and set the most significant bit on in the second "\$".

Much of what I write is still in Z80 assembler language and so ZSID is an important tool for me, giving Zilog mnemonics and full Z80 tracing. However, the format of the memory dump command in ZSID has always irritated me. Hex and ASCII data are displayed on alternate lines which limits the screen width required (my guess is that the author of ZSID only had a 64 column screen). This means that only half as much memory can be displayed at a time compared with the display format used by DDT where hex and ASCII are on the same line.

I recently got around to patching ZSID to give the same memory dump screen as DDT. Only five bytes are involved as outlined in listing two. Before making these patches to ZSID examine listing three, an example of the results, to determine whether you want to proceed. I was concerned not to cause problems with the relocation technique used by Digital Research so when patching out calls to subroutines rather than substituting NOPs which might then be modified I simply made the call address that of a return instruction.

Thanks Neil! Since DRI no longer supports CP/M utilities, patches like these are very important. If you have something useful please send it along. Talk to you next issue. (u)



Z SETS YOU FREE!

Z Operating System, an 8-bit OS that flies! Optimized HD64180/Z80 assembly language code — full software development system with proven linkable libraries of productive subroutines — relocating (ROM and RAM) macro assembler, linker, librarian, cross-reference table generator, debuggers, translators and disassemblers — ready to free you!

High performance and flexibility! Productivity results from dynamically customized OS environments, matching operator, tasks and machine.

Real-time control kernel option allows quick software development for industrial control applications, other tools and utilities for office desk-top personal computing functions, local area networks to Ethernet, AppleTalk, Omninet, ArcNet, PC-Net (Sytek) — from micro to mainframe command, control and communications. Distributed processing application programs are easily developed.

- Extreme organizational flexibility, each directory another environment
- Multiple Commands per line
- Aliases (complex series of commands known by simple names) with variable passing
- Named Directories with absolute password security
- Full-screen command line editing with previous command recall and execution
- Shells and Menu Generators, with shell variables
- Command-file search Paths, dynamically alterable
- Screen-oriented file manipulation and automatic archiving and backup
- 512 megabyte file sizes, 8 gigabyte disks handled
- Auto disk reset when changing floppies
- TCAP database handles characteristics of over 50 computers and terminals, more easily added
- Tree-structured online help and documentation subsystem
- 76 syntax-compatible support utilities

Your missing link has been found — Z! Now fly with eagles! Fast response, efficient resource utilization, link to rest of computing world — shop floor to executive suite, micro to corporate mainframe. Call 415/948-3820 for literature.



Echelon, Inc. 101 First Street • Suite 427 • Los Altos, CA 94022 • 415/948-3820

Listing 1 - SUBMIT.COM PATCH

Address	Old Data	New Data
05BB	24	A4

Listing 2 - ZSID PATCH

Address	Old Data	New Data	
125F	5F	BF	default number of bytes displayed-1
12C1	D1	85	was call to CR/LF subroutine
12C4	06	01	spaces between hex and ASCII
12E2	77	85	was a space between ASCII characters
12E5	77	85	was a space between ASCII characters

Listing 3 - ZSID PATCH RESULTS

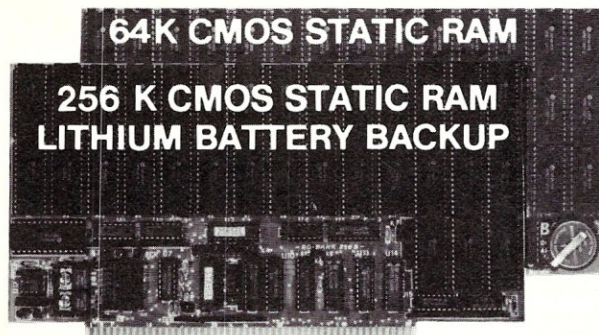
Before:

```
0100: 01 F9 21 C3 3D 01 43 4F 50 59 52 49 47 48 54 20
      . . . . . C O P Y R I G H T
0110: 28 43 29 20 31 39 37 37 2C 20 44 49 47 49 54 41
      ( C ) 1 9 7 7 , D I G I T A
0120: 4C 20 52 45 53 45 41 52 43 48 20 20 20 20 5A
      L R E S E A R C H Z
```

After:

```
0100: 01 F9 21 C3 3D 01 43 4F 50 59 52 49 47 48 54 20 . . . . . C O P Y R I G H T
0110: 28 43 29 20 31 39 37 37 2C 20 44 49 47 49 54 41 ( C ) 1 9 7 7 , D I G I T A
0120: 4C 20 52 45 53 45 41 52 43 48 20 20 20 20 5A L R E S E A R C H Z
```


HIGH PERFORMANCE RAM



✓ COMPARE

	8/16 BIT DATA	Bank Selection MPM, OASIS, CROMIX-D	10 Megahertz Speed	Software Write Protect	Battery Backup Option
Compupro Ram 22	✓	NO	✓	NO	NO
Octagon 256K	✓	NO	✓	NO	NO
Cromemco 256KZ II	✓	NO	NO	NO	NO
Dynamic Boards	✓	✓	NO	NO	NO
BG-Bank 256S	✓	✓	✓	✓	✓

GUARANTEED IN YOUR SYSTEM CROMIX-D • MPM • CCS • OASIS • AMOS

✓ **PLUS:** 8/16 BIT TRANSFERS • 24-BIT EX. ADDRESSING
8-12 MHZ • 2K DESELECTS • RAM-EPROM MIX
IEEE 696/S-100 • LOW POWER • FULLY STATIC

LITHIUM BATTERY BACKUP avoids power failure crashes intelligently. Unique POWER-FAIL-SENSE circuit allows processor to save register information and disable board before POWER FAILURE CRASHES memory.

BG BANK 256S..... ~~\$295~~ Battery Backup ~~\$70~~
BG BANK 64S..... ~~\$290~~ Battery Backup ~~\$50~~

call for new lower prices
MEGABYTE MADNESS
FOUR 256S CARDS
\$2295 call for new lower prices

BG COMPUTER APPLICATIONS, 206 Brookside,
Bryan, Texas 77801. International orders add 30%.
(409) 775-5009

Pascal Tutorial

Introducing a terrific new tutorial you can use to learn to program in the modern programming language Pascal. Pascal uses structured methods for program definition leading to an easy to write, and easy to understand and maintain program.

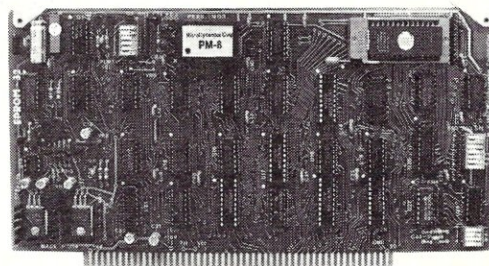
- If you are not a Pascal programmer, this tutorial will clearly describe the most fundamental aspects of Pascal programming. It assumes you know nothing about programming and leads you step by step from the fundamentals to the advanced features of Pascal.
- If you are already somewhat experienced in Pascal programming, this tutorial will lead you into using those aspects of Pascal that you have been ignoring. A clear definition of records, pointers, dynamic allocation, and even linked lists are given along with examples of each.
- There are 71 Pascal source files on a disk that you compile and run after examining them using the explanations given in the preprinted instruction manual.
- This tutorial is especially written for TURBO Pascal 2.0 or 3.0 but could be used with any Pascal compiler since the majority of the programs are written in standard Pascal as defined by Niklaus Wirth.
- The last chapter includes several example programs to illustrate program development methods. These programs are also on the disk and can be compiled and run as further examples of programming in Pascal.
- Requires MS-DOS or PC-DOS 2X or later and can be run on any IBM-PC or compatible computer. Equipment requirements are only as needed for your compiler.
- Not Copy-Protected and guaranteed to be effective. If you are not satisfied, we will gladly refund your money if requested within 30 days.
- Sent postpaid anywhere in the USA. Send check or money order for \$24.50 to:

Coronado Enterprises
12501 Coronado Ave NE
Albuquerque, New Mexico 87122

Phone (505) 293-5464 Evenings and Weekends

S-100 EPROM PROGRAMMER

EPROM-32



- Field-proven board meets IEEE-696 standard.
- Programs 1K through 32K (byte) EPROMs.
- Textool zero-insertion-force programming socket.
- EPROM is programmed through I/O ports and can be verified through I/O ports or located in memory space for verification.
- Programming voltage generated on-board.
- Personality Modules adapt board to EPROMs:
PM-1—2508, 2758 PM-3—2732, 2732A PM-6—68764
2516, 2716 PM-4—2564 PM-8—27128
PM-2—2532 PM-5—2764 PM-9—27256
- Feature-packed CP/M-compatible control software includes fast programming algorithm.
- One year warranty.

\$269.95*
(A & T)

MicroDynamics
Corporation



P.O. Box 15777 • Memphis, TN 38187
(901)-755-0619

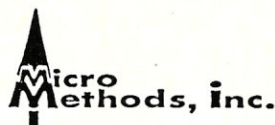
* Price includes EPROM-32, documentation and two Personality Modules (specify). Additional Modules—\$17.95. Control software on 8" SSD diskette—\$29.95. UPS ground—\$2.00. UPS air—\$4.00. COD—\$2.00. foreign add \$15.00. VISA & MASTERCARD welcome.

See Dec. 1983 **Microsystems** for a review of the EPROM-32.

PC/FOCAL

FOCAL® language interpreter

This is the introductory programming language made famous on the PDP-8. All the familiar commands, Ask, Type, Set, For, If, Do, and so on, are present, plus a whole host of new graphics commands, DOS and ROM access commands, and disk I/O commands that tailor PC/FOCAL uniquely to harness the full power of your IBM PC/XT or compatible. Puts fun back into programming; you can do so much with so little code. Easy to learn and use, on-line help files, decimal and hex console I/O, direct memory color graphics, access to registers, access to all ROM BIOS and DOS facilities, joystick, bit manipulation, 8087 support built-in. Useful example programs on disk, including hex disk editor entirely in PC/FOCAL listable on one page. PC/FOCAL disk with user manual \$49.95 US postpaid. Requires PC or MS-DOS 2.1 or higher. UPS shipping \$5 (nonUS shipping \$10)  



118 SW First St. - Box G
Warrenton, OR 97146
(503)861-1765

POWER USERS - PROGRAMMERS

Now, in one integrated utility, a super powerful DOS Shell and the best disk management tool available. **XPIP** provides powerful replacements and extensions for most DOS commands plus easy command recall and editing. Execute any DOS command or application program from within the **XPIP** Shell. Traverse the directory tree with single keystrokes. Create and access hidden files and directories. Easily edit any file on the disk, even .EXE files, in either HEX or ASCII. Complete online help plus either menu or command line operation. Too many features to describe in any size ad. Now includes a powerful file sort utility. Not copy protected. CP/M version also available. All at a "public domain" price of \$29.95 + \$3.00 s&h. You'll wonder how you got along without it. Order your copy of **XPIP** today.

System Facilities, Inc.

P.O. Box 7079 (M)	P.O. Box 5664 (M)
Charlottesville, VA 22906	Lacey, WA 98503
(804) 977-5245	(206) 456-0628

MasterCard and Visa Accepted
VA sls tx: \$1.20 WA sls tx: \$2.25



PUBLIC DOMAIN SOFTWARE

Our public domain software library:

- now in excess of 1000 disks
- over 30,000 programs
- includes the complete CP/M UG, SIG/M, PCSIG libraries
- 20-40 new volumes added per month
- supporting over 200 CP/M computers (including Northstar and Apple) and all major IBM type computers (including DEC)

Available In One of Two Ways

Via Modem

- online directly from one of our 11 remote RCP/M, RBBS or FIDO Host systems
- 24 hours per day, 7 days per week
- limited access, restricted membership
- access fee only \$35 Can (\$30 US) covers one year access
- supporting 300/1200 and 2400 baud modems
- telenet access also available

By Mail

- all software available by mail on over 200 different disk formats
- priced by disk, from \$10 Can (\$8 US) per disk
- all major user groups available

Canadian Users

- a full line of commercial software available
- 12 different brands of modems including the US Robotics 300/1200 baud Password and the US Robotics 300/1200/2400 baud Courier
- Echelon and Night Owl software available directly from us



For a free minicatalog,
call (416) 239-2835 or write:

CANADA REMOTE SYSTEMS LTD.

4691 Dundas Street West
Islington, Ontario, Canada
M9A 1A7

Phoenix' PFIx-Plus Debugger

by Randy Davis

Phoenix Computer Products Corporation has long been known for its professional software utilities for MS-DOS machines. Such Phoenix utilities as PLIB86 and PLINK86 start with capabilities already included in DOS, and add significant features to bring them up to the level expected by professional programmers. In similar fashion, PFIx86-plus starts with, but goes far beyond, the simple capabilities of the DOS supplied DEBUG utility.

The Phoenix debugger is available in two different forms, the standard PFIx86 and the enhanced PFIx86-plus. The two debuggers are identical, even sharing the same manual, except that the -plus version includes the ability to read and use a symbol table. Rather than confronting the user strictly with hexadecimal addresses as DEBUG does, PFIx-plus replaces hex addresses with labels wherever it can. This aids tremendously in understanding the disassembled listing appearing on the monitor. I will only be discussing the enhanced -plus debugger, although everything said, except for references to program labels, applies equally well to the standard version.

THE PACKAGE

PFIx86-plus comes in an attractive 3-ring hard back binder, roughly the same size as the DOS manuals. Enclosed within the manual are the standard registration papers and a single system diskette. The shrink wrap license is the standard *one machine* agreement. Thankfully the PFIx system disk, including the debugger itself, is not copy protected. Included on the system disk is an example configuration file, some added utilities and a READ.ME file containing the most recent errata, including errors in the PFIx manual.

Along with both a table of contents and an index, the manual contains well written chapters dedicated to installation of the software, principles of operation, and the meaning of the various debugger commands. Due to the profusion of debugger commands and the insistence on the manual's part of explaining separately each and every one, the manual seems much larger and cumbersome than it needs to be. How-

ever, appendices containing debugger command summaries also appear at the end of the manual for quick reference. I had difficulty with the tutorial chapter since it did not seem to completely match the example program on the system disk.

One other problem: I did not feel that Phoenix spent sufficient time describing the use of their powerful breakpoint facility. Further, the configuration file includes the ability to adjust some rather critical parameters; however, without more information, few will venture to try it.

THE PFIx DEBUGGER

PFIx consumes roughly 60k of system memory, not a large appetite by today's standards. The debugger appears to write to the screen using direct screen I/O. Although this necessitates highly IBM compatible hardware, screen updates become practically instantaneous. PFIx will work with two displays, if present, by using one as the debug screen and the other as the user program screen. Although this works very well, PFIx works almost as well on a single monitor system by saving off the user screen when at a breakpoint and maintaining the debugger display in the background when the user code has control.

The PFIx debugger itself consists of two files: the PFIxPLUS.EXE and an ASCII configuration file called PFIx.CFG. By editing the .CFG file the user has some flexibility in configuring the debugger for his particular desires. For example, the NEC V20 in my PC confused PFIx into thinking that it was running on an 80286 processor, like that in the AT. A quick edit of PFIx.CFG forced the processor type to 80186, the available option most similar to the V20 in instruction set.

Commands are entered on a command line, very similar to that made popular by Lotus in 1-2-3. The sliding inverse video bar provides a form of on-line help, making PFIx almost self explanatory. To facilitate quick entry, the most common commands are accessible by the function and Alt keys. This is fortunate as the many available options often make it necessary to traverse several levels of menu before the desired command may be completed.

There is one unfortunate difference between PFIx' command input and that of Lotus. Once a command has been completed, PFIx immediately puts you back at the *root* level, rather than leaving you at the *current* level. This often necessitates needlessly climbing back down the same command path multiple times.

One of the primary areas of any debugger is that of breakpoints. PFIx allows for up to 10 temporary and 10 permanent breakpoints. Temporary breakpoints are like those found in DEBUG, all breakpoints being removed when any one of them is encountered. Permanent breakpoints remain, however, until specifically removed. Permanent breakpoints may be enabled or disabled. In addition, permanent breakpoints can have loop counts and may be conditional. For example, the user might specify to stop on the fifth time that a certain instruction is executed with the AX register set to 0.

PFIx understands logical operators. This makes it possible to write complex conditions, such as AX=0 and BX≠0, or stopping for the result of a particular calculation that has a particular value. Apparently, the programmer can get quite creative in this area. Since the manual does not go into great detail, I do not know what the limits of complexity of the conditional might be.

Permanent breakpoints need not break to the debugger screen when encountered; other options are to call a user provided routine, enable another breakpoint or disable another breakpoint. As an example of this type, suppose that I had a function which gave problems whenever it was passed an input value of 0. I could set a breakpoint at the point where the function messes up, and disable it. I would then set a second breakpoint at the beginning of the function, which would enable the first breakpoint if it saw an input value of 0. This type of *cascading* of breakpoints is very helpful in locating bugs which only occur infrequently.

An unusual type of breakpoint, which Phoenix calls a global breakpoint, may be specified. Global breakpoints do not specify an instruction location, only a count and a condition. When such breakpoints exist,

PFix single steps the user's program, checking for the condition after each instruction. For example, suppose I know that a particular memory location is being cleared erroneously, but I don't know from where. I can set a global breakpoint "proceed until location xxxx is zero". My program will execute one instruction at a time with PFix keeping a constant vigil on that location. Even though my program executes much slower this way than it otherwise would, this method is hundreds of times faster than if I must manually single step the program, while watching the location myself.

One of PFix' greatest powers lies in its ability to manipulate symbols (as mentioned earlier, here we are only discussing PFix-plus). A program linked with Phoenix' PLINK86 (not included) can optionally contain a symbol table within the executable .EXE file. Bring up such a program from PFix, and the symbol table is loaded automatically. No longer must I specify an address in the form, 1234:5678; I can now specify CODE_SEG.MY_PROG, or any other label from within my program, and the debugger knows exactly what I mean. Further, by referencing such symbols, breakpoints may be set in overlays, even if the overlay containing the breakpoint address is not currently loaded into memory.

Much to Phoenix' credit, PFix does not require the use of PLINK86 to take advantage of its symbolic capabilities. The PFix system disk includes a utility, ADDSYMS, which can take the .MAP output file from the standard DOS linker and combine it with the .EXE program file. When brought up under the debugger, PFix contains every symbol which appeared in the linker's map file. Although such a combined file can also be executed normally from DOS (which will ignore the symbol information), a second utility, SYMTABLE, has been provided to strip the symbol table from a working program without the need to relink. (Note that those who use the PLINK86 linker need only include the /SYMBOLS switch to include symbolic information in the .EXE file.)

Once in the debugger, symbols may be added (or removed) manually. For example, once I have narrowed a problem down to a particular area, I might set a label, say TROUBLE_SPOT, at that location. With this label I can easily return to the location of the problem. PFix will not write the edited symbol table back out to the .EXE file, but it will write a separate symbol table file, which can subsequently be reloaded. This ability to create labels is also a great aid when debugging code to which I do not have the source.

When stepping through a program, PFix gives the operator two options, trace and single step. Single step works exactly like the T command in DEBUG, executing

the next single instruction (actually the number of instructions is adjustable and need not be 1) and then returning control to the debug screen. Often times, however, the user would like to stay within the current routine, treating a call to a subroutine or a system call to PC-DOS as one instruction, not halting until after control has been returned to after the call. The PFix trace feature offers just this option. Trace also works for the instructions which perform more than one function (such as REP MOV, PUSH, etc.). For convenience, the trace and single-step, as well as the set temporary breakpoint, are accessible via special function keys.

When using PFix as a debugger, you can compile and link your programs normally. However, since neither the Phoenix PLINK86 linker nor the PC-DOS linker include symbols unless told to, you will at least need to add a few switches to your link step to fully utilize the symbolic features. Many compilers will also require switches during the compilation step to add symbol information to their output.

Unfortunately, Turbo Pascal will not generate a .EXE executable file and cannot include symbol information. C defines that all procedures declared are automatically public, unless specifically marked otherwise; therefore, C symbols are included automatically. Assembler source routines should have a PUBLIC directive for each symbol which the user wants PFix to know about. Many compilers will optionally include source code line numbers as symbols (Lattice C compiler generates such symbols if the -d switch is used). The documentation for each compiler will explain how to include symbol information in the object file.

The DOS linker must also be persuaded to include this information in the executable file. First, the default is for the linker to not generate any .MAP file. This must be overridden by providing a name to the prompt "Map file [NUL]:". Adding the /Map (or /M) switch to the link command includes procedure names in the .MAP file. Add the /Line or (/L) switch for inclusion of line numbers. Once generated, the .MAP file information is included into the .EXE file using the Phoenix supplied utility ADDSYMS mentioned earlier.

CONCLUSION

I used PFix for several weeks before beginning this review to familiarize myself with its features. Being something of a hacker, I had several programs lying around, waiting for those last bugs to be driven from them. One assembler program, in particular, had a bug which had already eluded many hours of assault from DEBUG. After reassembling, with PUBLICs inserted for the main symbols I needed, and relinking with the /M switch, I

applied PFix to the miscreant program. The bug had been found and permanently banished from my program within the hour!

Overall I found PFix to be an excellent program debugging tool. It provided all the abilities one would expect of a professional package. It never crashed even when performing some pretty tricky stunts. PFix puts a tremendous amount of debugger power at the user's disposal. Even though the layout of the special function keys seemed completely arbitrary to me and I have yet to be able to remember them, this seems like a picky point. I guarantee that anyone who uses the PFix-plus symbolic debugger will never return to simple DEBUG. μ

Mail

continued from page 26

interesting CP/M computer because of the sound and graphics capabilities. I had no problem using the sound and graphics capabilities of the C-64 from CP/M. The C-128 added another video chip that emulates the IBM-PC color display abilities. Very little software takes advantage of it yet. Currently, the C-128 has 128K in CP/M mode. Commodore is planning to produce a memory expansion cartridge that can be used as a RAM disk in CP/M mode.

Respectfully,
Glynn E. Stafford, Jr.
Waldorf, MD

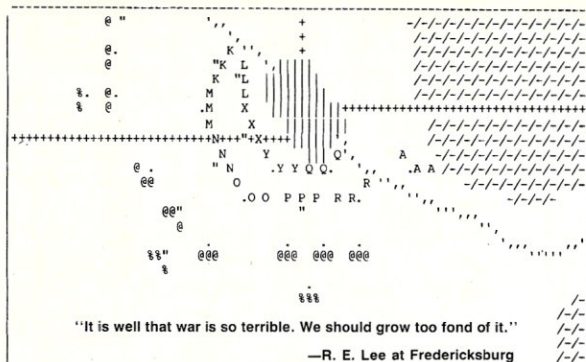
FUTURE ARTICLE TOPICS

Dear Sol,

I am glad to see the magazine is still surviving; I hope it grows until it positively flourishes. There are still a lot of things I would like to see. For a start, the series of graphics standards that the old rag was running when it folded was a Good Thing. It is a difficult area to get into unless you have someone fresh out of college to help or are in a big firm that can afford to make mistakes. I fear half your problem would be finding people capable of writing this type of article. Information on Concurrent in all its forms is welcome as well. It is becoming the multiuser operating system and is very popular over here (40% of DR's business is outside the States!).

I suspect that a large part of your readership are professional developers; the conditions that spawned the hackers have disappeared in the new era of regimentation. But, they are certainly enthusiastic professionals, and the options for producing more polished and professional systems are opening out fast. I personally am trying to decide whether to go for GEM or Windows. A series of articles comparing development with each of these and with Topview would be fascinating. Hints as to what works well for the user would be as interesting as technical information on using the toolkits.

Justin Smith
London, England μ



Robert E. Lee would have been very fond of **GENRAL**. **GENRAL** simulates the anxiety, uncertainty, and despair felt by the commanders of armies in an imaginary war taking place a century or so ago. **GENRAL** is . . .

. . . a **TWO-PLAYER** game. It requires **TWO** CP/M computers (Z80 or 8080) communicating via modem or cable at 300 baud. The computers are non-playing "umpires"; they produce statistical reports and animated displays of the battle situation, and relay the players' commands between the machines.

. . . a **REAL-TIME** game. It is always "your turn." The video display changes continually as time passes.

. . . a **COMPLEX** game. Each player controls an army of twenty-six units. There can be action in several places at once, and the decision which to attend to first can be critical.

. . . a **SERIOUS** game. It has had over five years of development and testing. A typical game lasts about two hours and requires constant, intense concentration. The game has an immense variety of possible situations, and does not grow stale with repeated playing.

GENRAL is available from: Tee-Kay Software
P.O. Box 23771
Columbus, Ohio 43223

Diskette and User's Manual: \$50.00
User's Manual only: \$10.00
Serial I/O Software Listings: \$ 2.00

A variety of 5" disk formats is available (including Apple), as well as standard 8" SSSD. Please specify desired format when ordering. NOTE: A small amount of customization of screen and modem control software may be required. Sample ASM sources (8080 code) are included on the **GENRAL** diskette. Listings are available on request.

QPARSER™

Translator Writing System

THE PRODUCTIVITY TOOL FOR SOFTWARE DEVELOPERS

QPARSER assists you in writing:

- * Compilers * Translators * Interpreters *
- * Prototypes * Simulators * Syntax Checkers *
- * Data Converters * Assemblers *

QPARSER is a unique LALR(1) parser generator:

Generates complete source code for your application in C, Pascal, or another language of your choice; Extensive examples include a Pascal subset compiler, assembler, and simulator;

The widely used college text, *Compiler Construction: Theory & Practice*, from SRA Associates, was written by the author of QPARSER

Lauded by both industrial and university users

Available for: IBM PC,XT,AT; DEC VAX; HP 9816; MACINTOSH
(PC System \$400; Demo \$10; Educational/Site Licenses available)

"LEADERS IN SOFTWARE TOOLS"

QCAD
SYSTEMS, INC.

1164 Hyde Ave., San Jose CA 95129

Toll-free Orders: (800) 538-9787; In CA: (408) 727-6671

PORT-A-SOFT

Aardvark to Zorba

We Can Handle It



**TRANSFER
PROGRAMS, DATA AND
OTHER FILES BETWEEN
OVER 400 COMPUTERS!**

13 OPERATING SYSTEMS

WORD PROCESSORS TOO!!

Prices From \$7.00 Per Disk
Software and Hardware So You Can Do Your Own

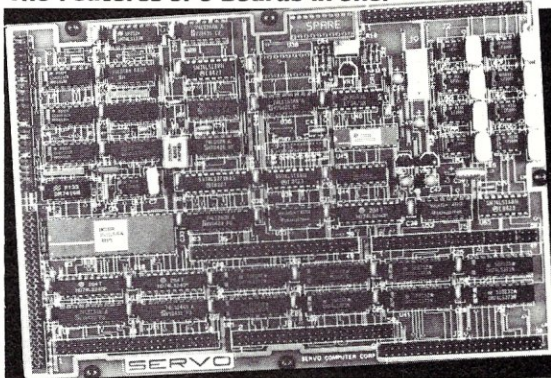
**Call or write today for your
FREE CATALOG**

P.O. BOX 1685 555 S. STATE, STE. 12
OREM, UT 84058 (801) 226-6704

SERVO INDUSTRIAL PROCESS CONTROL BOARD

**14 Bit Lab Accuracy
Industrial Durability**

The Features of 5 Boards in One.



The Servo Process Control Board is a general purpose multi-channel analog and digital input/output controller which can communicate with and control most medium-scale industrial processes. Designed to work with the Servo Industrial Microcomputer, (or other computers which support the SASI interface), it offers great accuracy at prices starting as low as \$495.

■ American made ■ MILSPEC components ■ 1 year warranty

Maximum configuration includes: ■ 192 analog inputs ■ 64 analog outputs ■ 128 digital inputs ■ 512 digital outputs

■ Based on the INTERSIL 14 Bit A to D converter ■ Communicates with any micro or mainframe computer ■ Easily programmed in high level languages (Turbo Pascal, Microsoft Basic, Fortran) ■ PID, FFT, Thermocouple linearization and laboratory DAS software supplied. ■ Custom software and hardware engineering assistance available.

SERVO
SERVO COMPUTER CORPORATION

360-B N. Ellensburg St.
Box 566
Gold Beach, OR. 97444
(503) 247-2021

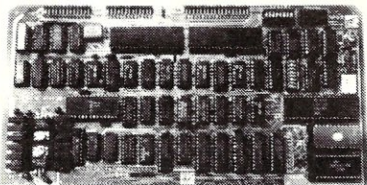
Quality S-100 Products

Now Available! FULCRUM'S New...
256K STATIC RAM  8/16 Bit Transfers
24 Bit Extended Addressing Battery Back-Up **\$495**

Designed For Speed & Flexibility.

MPUZ CPU

This NEW MPUZ CPU utilizes the Z-80 8MHz uP as a basis for its 8MHz CPU for S-100 systems, and has been carefully designed to meet the requirements of the IEEE - 696 standard. The quality and performance this CPU provides is rarely found in S-100 products, and you can see why... **only \$299**

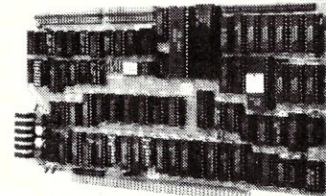


- ▶ 4 or 8MHz clock rate
- ▶ Two RS-232 serial ports
- ▶ Centronics printer ports
- ▶ Real time clock with battery back-up
- ▶ Vectored interrupts to any block location in memory
- ▶ Programmable timer
- ▶ ROM monitor
- ▶ Power on Jump
- ▶ On board wait states
- ▶ 2K of RAM space
- ▶ 24-bit extended addressing
- ▶ Latched Status
- ▶ Front panel compatibility
- ▶ MPM support

Best Value In Disk Controllers.

OMNIDISK

Now the FULCRUM OMNIDISK offers S-100 systems users a unique marriage of component compatibility and technological innovation. These together produce features not found in any conventional disk controllers made today. See for yourself what tomorrow looks like... **only \$299**



- ▶ Simultaneous support of both 5 1/4" and 8" floppy disks and hard disks
- ▶ Complete 24 bit DMA
- ▶ Power on boot for 5 1/4" and 8" floppy and hard disks
- ▶ Power on boot PROM
- ▶ On board de-blocking to save RAM space over BIOS
- ▶ Interfaces with the WD 1001* hard disk controller
- ▶ Supports 13 devices simultaneously
- ▶ Full track buffer allows controller to recall entire track
- ▶ DMA'S at 10 MHz
- ▶ Supports MS DOS
- ▶ 10K on board buffer saves two K of TPA

So before you buy another S-100 component, call or write for our FREE catalog. And see how your system can benefit from the FULCRUM difference.

*CP/M *2.2 configured for OMNIDISK \$60. *Trade mark of Digital Research. FREE U.P.S. ground shipping on prepaid orders. Shipping is added to VISA, M/C. and C.O.D. orders. CA residents, please add sales tax.



707/433-0202

**FULCRUM
COMPUTER PRODUCTS**

459 Allan Court, Healdsburg, CA 95448.

Also in FULCRUM'S Family: OMNIRAM 64K memory board. Serial I/O 2-2 & Video I/O Interface boards. Relay board. I-8080, 8015 and 8035 main frames with 21 slot mother boards. CPA front panel and A/D board.

The Holliston Challenger XT-186

by Charles H. Strom

Leafing randomly through the pages of any of today's microcomputer-oriented periodicals, one is faced with a plethora of IBM PC-compatible system boards. Many strive to be just like IBM; others add a feature or two such as 640Kb of RAM on the board, dual speed operation, etc. A few have gone a step further in an attempt to attain performance improvements while minimizing incompatibilities with the existing base of IBM-specific software. One such board is the Challenger XT-186, made by Holliston Computer, Box 615, Chepachet, R.I. 02814 (401-568-0522). We had the opportunity to use this board for several weeks and have prepared this report on its performance.

THE LAYOUT

The Challenger is dimensionally identical to the IBM-XT system board, so it slips in easily. We have been testing the board in a clone system consisting of the Challenger, a monochrome video card, floppy disk controller, DTC hard disk controller with a CMI 6426 drive, and a serial/clock-calendar board. The system board is based on an 80186 running at 8MHz (10MHz is an extra-cost option) with 640K of parity-checked RAM taking advantage of the true 16-bit wide data path afforded by the CPU. There are eight expansion slots available, six of which are PC-compatible. The two additional slots are quasi-AT compatible. Not all of the address lines are brought out to the connector and Holliston is currently investigating the usability of various AT compatible extended memory boards. We did not have an opportunity to use the extended slots, but the PC-type slots appear to be pure clones. There is no 8087 socket; however, a small piggy-back board is available.

The XT-186 has two sockets for system ROM; since it employs a true 16-bit architecture, it is not possible to substitute IBM or clone ROMs. Fortunately, virtually no problems with the XT-186 BIOS were experienced. With a few exceptions noted below, all software ran flawlessly. An optional ROM has recently been announced by Holliston. It supports the IBM AT combination floppy/hard disk controller and thus allows

the use of 1.2Mb floppies as well as two hard disks. Unfortunately, we did not have the opportunity to test this ROM.

PERFORMANCE

The best way of assessing the utility of the board is to put it to everyday use. The Norton Utilities system benchmark reports speed equivalent to 4.2 PC's. A more meaningful benchmark is the famous Sieve of Eratosthenes. Paul Homchick, our co-sysop on the GENIE IBM Roundtable, created a stand-alone program that finds 1899 primes and displays the elapsed time. Here are a few representative results of this compute-intensive benchmark relative to an XT:

IBM PC-XT	1.00
Compaq	1.65
CompuPro (8088)	1.78
AT&T 6300	2.10
AT&T 6300 (V30)	2.62
IBM PC-AT	3.19
Holliston XT-186	3.07

Clearly, the XT-186 is a respectable machine in the speed department. The only issue remaining is compatibility. Both MS-DOS 2.1, and 3.1 for the IBM PC, boot up with no difficulty at all; virtually all *well-behaved* applications and utilities perform correctly. This includes classics such as WordStar, DBase III (Holliston reports a problem with DBase III Plus which they are investigating), etc., all of the MS-DOS utilities, and the dozens of public domain and shareware programs we have used while we have had custody of the XT-186.

It is much simpler a matter to describe the incompatibilities as they are few in number. Several programs, such as CopyWrite by Quaid and earlier versions of Media Master (a format translator) talk directly to the floppy disk controller and fail with the XT-186. Holliston explains that this is caused by too rapid an execution of closely-spaced I/O instructions (due to the high speed and instruction caching of the 80186) causing overrun on the peripheral device. The matter is simple to resolve if source code is available, but that of course is not the case in the above-mentioned examples. Intersecting Concepts has updated Media Master to run

on an AT. This enhancement also allows proper operation on the XT-186. CopyWrite is written specifically for the PC and XT and not the AT. Therefore, without provisions to insert software delays at strategic places, CopyWrite will not run correctly. It would be useful to include a hardware or software switch to allow a slow mode of operation for such extremely hardware-dependent applications, but there is no such feature on the XT-186. If a package is designed to run on an AT, there should be no timing problems on the Holliston board.

Another incompatibility we discovered is with Digital Research's Concurrent PC-DOS version 4.1; attempting a boot yields a memory parity error and subsequent program halt. This is unfortunate, as we feel that CPCDOS is a natural for high speed operation. The overhead of multiuser, multitasking operation afforded by CPCDOS leads to slow operation on a stock PC. We use this operating system routinely with a 9.5MHz 8086-based accelerator card and a similar, acceptable system speed would be experienced with the Holliston board as well. We seem to be in the minority in our fondness for CPCDOS; in fact our mere mention of it is often met with raised eyebrows by the bulk of PC users. Thus, this incompatibility should not be a critical one for most users.

THE BOTTOM LINE

Prices for the XT-186 are reasonable. As of this writing, an 8MHz system board populated with the maximum of 640K of RAM lists at \$595, or \$495 without memory. A 10MHz system costs an additional \$150 (well worth the incremental expense in view of the 25% speed improvement). The AT-compatible ROM is an option priced at \$50. The cost of the 8087 piggy-back board has not been firmed-up as of press time.

In summary, the Holliston Challenger XT-186 is a board to consider if you want to upgrade your PC or XT system to AT performance. With the same form factor as the PC/XT boards, its an easy replacement providing superior performance with minimal sacrifice of compatibility and does not require any change of software or operating habits. [E]

**FOUR TIMES
FASTER**

The XT-186 Mother Board

Fully IBM PC Compatible

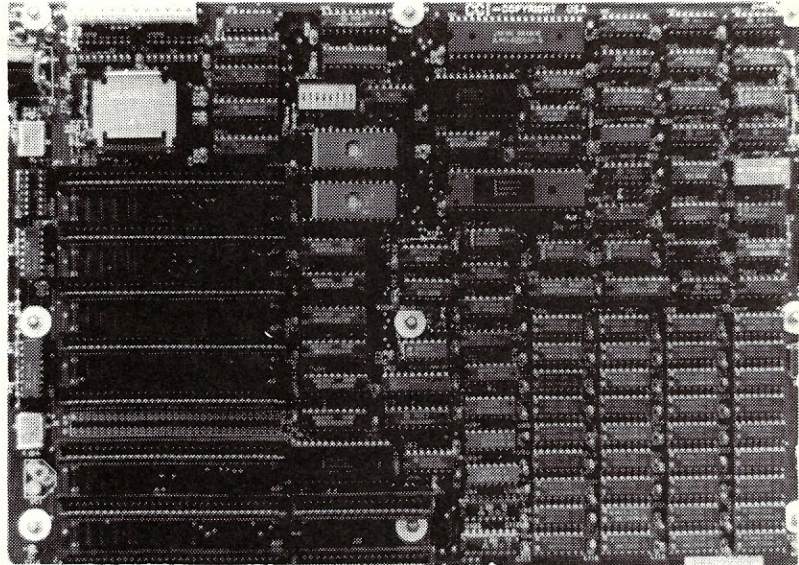
**EXACT SIZE
REPLACEMENT**

STANDARD FEATURES:

- Intel 80186 CPU, 8 MHz
- On-board 640K, parity checked RAM memory
- 8 slots for I/O Interface Board
 - 6 - 8 Bit Data
 - 2 - 16 Bit Data
- 4 channel 8237 DMA
- 8 channel 8259 Interrupt
- 3 channel 8253 Timer
- IBM PC-AT-like Bus with true 16 bit data path

OPTIONAL FEATURES:

- 10 MHz operation
- 8087 Numerical Processor Support
- ROM BIOS Version 2.0



The Challenger XT-186 offers full IBM PC-XT compatibility. It supports industry standard operating systems: MS-DOS, PC-DOS, 1.0 through 3.1, CP/M and runs all existing software with higher performance. The XT-186 mother board can directly replace any existing PC or PC-XT mother board, as it has the same physical dimensions and mounting holes. The Intel 80186 and 640K on-board 16 bit path RAM makes the XT-186 the fastest PC currently available.

MOTHER BOARD, 8 MHz, 640K, RAM, 8 I/O slots

NEW LOWER PRICE \$595.00

The XT-186 ROM BIOS version 2.0 supports the IBM PC-AT combined Hard Disk/Floppy Controller on the Challenger XT-186 mother board. With the new ROM BIOS installed the XT-186 can support two 1.2 MB floppy drives and two hard disk drives with the same fast access time of the IBM PC-AT. No other XT compatible system can offer either the speed or the disk storage capacity, (230 Mbytes maximum) of the XT-186.

XT-186 ROM BIOS version 2.0 Upgrade Kit

\$50.00

VISA and MASTERCARD Accepted |— Please Add \$3.00 Shipping

HOLLISTON COMPUTER

P.O. Box 615, Chepachet, R.I. 02814

401-568-0522

IBM PC-XT-AT, PC-DOS are trademarks of International Business Machines Corp., MS-DOS is a registered trademark of Micro-Soft Corp. CP/M is a registered trademark of Digital Research, Inc.

NOW AVAILABLE

XT-88 Turbo Board

A reliable, low cost mother board with IBM PC-XT compatibility, running 1.8 times faster. 8MHz hardware selectable to 4.7 MHz. On-board 8087 socket ready for co-processor chip installation. The 8088-2 processor with 8 I/O slots, 640K RAM and compatible ROM BIOS. . . .

\$299

No RAM **\$199**

AT-286 Mother Board

Here's the power you need to get the job done! IBM PC-AT compatibility enables you to run a broad range of programs, both personal and professional. The board comes with a built-in Real-time clock and 80287 socket for future co-processor insertion. The AT-286 with the Intel 80286 microprocessor running at 8MHz with 640K RAM and Phoenix BIOS processes information faster and more efficiently. Build a high-performance computer and boost your productivity.

Mother Board, 8MHz clock rate, 640K RAM, 8 I/O slots, Phoenix ROM BIOS **\$950**

No RAM **\$850**

Program Interfacing To MS-DOS

by William G. Wong

This article presents a real example of a loadable DOS device driver using the information described in the previous articles. The driver is for a memory disk (often referred to as *virtual* disk or RAM disk) which is a block device. The next article will present a character device driver.

A memory disk turns out to be a good example because it is readily implemented on any DOS system. A memory disk uses the processor's main memory to emulate a disk drive. The advantage is its transfer rate which is 10 to 100 times faster than a hard disk. The disadvantage, typically, is that this disk is volatile, and it uses up the memory resources which may be required by application programs. Both disadvantages can be overcome using hardware support, such as battery backup power supplies, and separate memory boards that can extend the memory systems to as much as eight megabytes of storage. However, specific support for these devices will not be presented here.

Macros were not used for portability. The source was assembled using the Microsoft Macro Assembler but any 8086/8 assembler should work. The object file was linked and converted to a .COM file format program. The origin of the program is 0 as opposed to 100 hex for normal .COM files. The file type of the resulting executable file was named MDISK.SYS.

The MDISK.SYS file is included on the boot disk and the following line of text is included in CONFIG.SYS, also on the boot disk.

```
DEVICE = MDISK.SYS
```

That is all that is required to use the memory disk drive on a DOS 2.x system.

The driver source is included at the end of the article and is referenced throughout. The source code is divided into four sections. The first contains device independent constants and definitions including offsets to various structures and values which may be placed into these structures. The second section defines common structures and variables used by all device drivers. The BIOS Parameter Block (BPB) structure is common to block devices although it contains device specific values. The third section contains source code which is common to all device drivers. The first three sections can easily be extracted from the memory disk source

Part VIII A Memory Disk Device Driver

and integrated with support code for other devices. In fact, these three sections will be used in the next article which supports a character device. The fourth section contains the code specific to the memory disk. It also includes the device specific variables and structures except the BPB which is in section two.

The assembly code is structured using CALL and RET instructions to access and terminate functions. Any jumps are within a function and implement conditional code segments or loops. Opcodes are normally positioned one tab position to the right of the left margin. An additional space is included if information is pushed on the stack. The space is not included after information on the stack has been removed. For example:

```
push    ax      ; save some registers
push    bx      ; on the stack
call    foo     ; do something
pop     bx      ; restore to registers
pop     cx
```

For those unfamiliar with the 8086 architecture, there are two types of call and return instructions, long and short. The long versions push 32 bit return pointers (offset and segment) on the stack and the short versions use only 16 bit offsets. Functions are bounded with special statements which define the type of call to use to access the function and the type of return instruction to use. For example:

```
sample  proc    far
        CALL    sample ; recursive call
        RET
sample  endp
```

The label *sample* is the address of the start of the function even though it appears at the end too. The words *far* or *near* indicate that calls to the function should use long and short values respectively. The CALL and RET instructions in this example would use 32 bit pointers.

1. DEFINITIONS

The first part of the program contains a set of constant definitions for offsets into the Request Header data structure and values for some fields. The following table of prefixes describes their usage.

Prefix	Usage
rh_	request header field offset
s_	rh_status field success value
e_	rh_status field error value
m_	rh_media field value
mc_	rh_check field value
a_	device driver header attribute field value

These offsets and values are used throughout the program. The offsets are defined with a leading constant plus the previous field offset. The constant is the size of the previous field in bytes.

Fields which are bit encoded may have multiple field values included as the value of the field. The device driver header attribute field is one example.

2. VARIABLES AND STRUCTURES

The variables and structure definitions come next in the source code. The very first item must be the first device header. There is only one in this example and the link field value is -1. Additional device headers would be linked via this field but the headers could appear anywhere within the program.

The next structure is the BIOS Parameter Block (BPB) table which is used for block devices only. There is one entry for each unit supported. The single entry is all that is required for this example. The offset refers to the BPB contained within the boot record image which comes next. This is used to initialize the memory drive and is retained because the BPB structure is used by the device driver.

The last variable is the address of the request header passed to the strategy routine. This is loaded by the interrupt routine for use by all support functions.

The last structure is the command dispatch table. It contains the offsets to the respective functions. The common part of the device interrupt routine uses this table to call the corresponding function based upon the command byte in the request header. Each entry in the table is a unique label even though the memory disk driver does not implement all functions. This

allows the table to be used for device drivers in general.

3. COMMON DRIVER SUPPORT CODE

The common device driver routines follow the constant and data definitions. The two driver entry points are included in this set of code. The device strategy routine (`dev_strategy`) is very short and simply stores the address of the request header where the interrupt routine can find it.

The device interrupt routine (`dev_interrupt`) saves, and later restores, the working register set on the stack. It sets up the data segment to match the code segment since many of the support functions operate best with this configuration. The request header address is then loaded and the command is used to index the command table. The corresponding entry is then used to call the appropriate function.

Each function returns the status code in the AX register whose value is placed into the status field of the request header before restoring the working register set.

All support functions simply return so they may be easily developed outside the device driver environment. The next section adds the device specific support code. All items up to this point can be used for any block device driver and, with minor changes, any character device driver.

4. MEMORY DRIVE SPECIFIC CODE

The device specific functions supported by the memory disk driver are:

Function	Description
<code>initialize</code>	setup memory disk
<code>media_check</code>	show non-removable disk
<code>build_bpb</code>	build BPB
<code>read</code>	read sectors
<code>write</code>	write sectors
<code>write_verify</code>	write and verify operation

Each of the supported functions will be addressed individually along with the two support functions `source_address` and `destination_address`. Non-supported functions all reference the function *unimplemented* near the end of the source listing.

The `source_address` and `destination_address` functions compute the segment and offset for the corresponding sector in the memory drive. They also compute the number of bytes to transfer based upon the number of sectors to transfer. The difference between the two is simply the segment and offset registers used. The `source_address` function uses registers DS:SI while the other uses ES:DI which corresponds to the 8086 string move register pairs. In essence the functions convert the parameters found in the request header block (sectors) into memory disk/8086 usable values (pointers and byte counts). The functions use the BPB values

It's 3 AM!



Do you know where your bugs are?

This C programmer is finding his bugs the hard way...one at a time. That's why it's taking so long. But there's an easier way. Use

PC-Lint

PC-Lint* analyzes your C programs (one or many modules) and uncovers glitches, bugs, quirks, and inconsistencies. It will catch subtle errors before they catch you. PC-Lint resembles the Lint that runs on the UNIX* O.S., but with more features and some awareness of the 8086 environment.

- Full K&R C
- Supports Multiple Modules—finds inconsistencies between declarations and use of functions and data across a set of modules comprising a program.
- Compares function arguments with the associated parameters and complains if there is a mismatch or too many or too few arguments.
- User-modifiable library description files for most major compilers.
- All warning and information messages may be turned on and off globally or locally (via command line and comments) so that messages can be tailored to your programming style.
- All command line information can be furnished indirectly via file(s) to automate testing.
- Use it to check existing programs, programs about to be exported or imported, as a preliminary to compilation, or prior to scaling up to a larger memory model.
- All one pass with an integrated pre-processor so it's very fast.
- Has numerous flags to support a wide variety of C's, memory models, and programming styles.
- Price: \$139.00 MC, VISA (Includes shipping and handling) PA residents add 6% sales tax. Outside USA add \$10.00
- Runs under MS-DOS* 2.0 and up, with a minimum of 128Kb of memory. It will use all the memory available.

Trademarks: PC-Lint (Gimpel Software), UNIX AT&T, MS-DOS (Microsoft).

NEW!! Amiga - Lint
Special Introductory Price
\$98.00

GIMPEL SOFTWARE

3207 Hogarth Lane • Collegeville, PA 19426
(215) 584-4261

to perform their computation so the size of the memory disk can be changed by simply altering the BPB values.

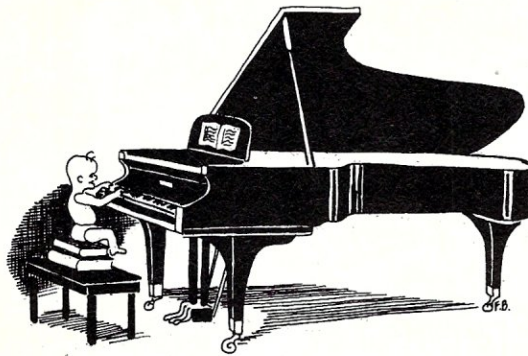
The `initialize` function is called once by DOS when the device driver is loaded. In theory, this code could be located after the end of the resident portion of the device driver because it is only used once. However, this is where the data for the memory disk is placed. Unfortunately, the first part of the memory disk contains data which is set up by the `initialize` function so it cannot overwrite itself. It is possible to move the code around but this increases the complexity of the system and would cause more confusion than it is worth. In fact, the code for this function is a very small percentage of the overall device driver size

since the resident portion of the memory disk driver includes the memory disk itself. It makes sense to put the initialization code at the end of the driver, if the memory disk itself were located on an extended memory board since the space could be returned to the system.

The first thing the `initialize` function does is compute the base segment of the memory disk. All sectors must be aligned to 8086 paragraph boundaries; this coincides nicely with the sector sizes that DOS uses, primarily 512 bytes. The base segment is saved in the device specific variable `mdisk_segment`. This is used by the `source_address` function to compute the proper segment for a given logical sector.

C-terp

The C Interpreter You Won't Outgrow



C-terp will grow with you as you progress from novice through professional to guru. Unbelievable, but true, the easiest-to-use C interpreter will provide you with the most advanced programming features for upward growth. Our exclusive **object module support** enables you to add libraries (like HALO, PANEL, Windows for C, etc., or your own homebrew libraries) to C-terp as you add them to your computing repertoire. Use C-terp as a microscope on your libraries! Flip a bit and allow our **software paging (NEW)** to handle those big jobs! There are no fixed-size tables to overflow, and C-terp can be configured for different screens and screen adapters (NEW). With multiple modules and **full K&R support**, we offer a dream C environment.

- Our new improved **configurable editor** competes with anything going.
- Speed -- Linking and semi-compilation are breathtakingly fast.
- Convenience -- Errors direct you back to the editor with the cursor set to the trouble spot.
- Symbolic Debugging -- Set breakpoints, single-step, and directly execute C expressions.
- Compatibility guaranteed -- batch file to link in your compiler's entire library. Supported compilers include: **Computer Innovations C86, Lattice C, Microsoft C 3.0, Mark Williams C86, and Aztec C.**
- Many more features including batch mode and 8087 support.

What Our Users/ Reviewers Are Saying

- "... easy to use, powerful, and a timesaver."
- "... we absolutely LOVE C-terp."
- "... has restored my faith in interpreters."
- "... a programmer's dream."
- "... wonderful technical assistance."
- "... increased our productivity by a factor of 40."
- "... the best C product ever, in any category."

- **Price: \$300.00 (Demo \$45.00)**
MC, VISA

Prices include documentation and shipping within U.S. PA residents add 6% sales tax. Specify compiler.

- C-terp runs on the IBM PC (or any BIOS compatible machine) under DOS 2.x and up with a suggested minimum of 256 Kb of memory. It can use all the memory available.

* C-terp is a trademark of Gimpel Software.

GIMPEL SOFTWARE

3207 Hogarth Lane • Collegeville, PA 19426

(215) 584-4261

The next operation performed by initialize is to copy the boot sector image to the first sector in the memory disk and then setup to the two File Allocation Tables (FATs). The FATs are identical and maintained by DOS. They are initialized so no files have been allocated. It is possible to set up pre-allocated files if required. These would also be referenced from the subsequent sectors which contain the directory. These are zeroed in this example, providing a completely empty disk with no allocated files.

The last thing the function does is to compute the address of the last byte used by the device driver. In this case it is the end of the memory disk which is well past the end of the program itself. This value is

computed by using the size of the disk in sectors which is found in the BPB. The value is placed in the proper request header field which DOS examines. The return value in the AX register indicates success.

Operations which the initialize function may perform in other block device drivers would include initializing a hard disk, determining a device's capabilities or current status, and setting the system to a known state. This would also be the place to set up interrupt vectors, stacks and interrupt routines.

The media-check function is very simple in this example because the device contains non-removable media, chips in this case. This is also true of most Winchester hard disks but would not be

true of floppy disks and removable disk cartridges. The value returned in the request header would depend upon whether the media had changed and what type of detection is provided to determine if the media had been removed. This may be tied into interrupt routines used by the driver (not the dev_interrupt routine).

The build_bpb function is used to generate the proper BPB for DOS. In this case, the operation is very simple because only one unit is supported and it cannot change. Also, the information is contained in the boot sector. Actually, the operation could be simpler by only using the resident BPB instead of reading the one from the boot sector. This operation was done to show how the function would be performed for more general block device drivers.

Dumb block device drivers would only have one BPB entry per unit because they only support one type of media. *Smart* block device drivers support various media types and usually determine the BPB based upon information from the device. This is typically a copy of the BPB in the boot sector. Although this is used for removable media devices, it can be used for non-removable media which would allow a common driver to be used for disks of different sizes or configurations.

The read, write and write_verify functions are very simple in a memory disk. They simply compute the memory address of the requested sectors and then transfer this data. The transfer is done using block move instructions at the byte level. This can be changed to moves at the word level since sector sizes are always a power of two. This would make a substantial difference if the processor is not an 8088 or 80188. This change could be implemented by simply dividing the number of bytes to transfer by two and replacing the byte transfer opcodes with word transfer opcodes. Also, the initialize function must be slightly altered to take into account the word count values being used.

The write_verify operation uses two tricks. First, the write function always saves the number of items transferred and leaves the registers pointing to the item just past the end of the buffer and the last sector in the memory disk. The 8086 string scan direction is reversed and the block compare instruction is used once the register have been adjusted accordingly.

MAKING CHANGES

The memory disk presented is set up for a reasonably sized drive which could be used for small programs, overlays or temporary files. It uses a small percentage of memory in a PC system with at least 512 Kbytes. However, the size is significant if used on a system with less memory.

The size of the disk can be changed by altering the values in the BPB. No changes are required in the rest of the code. In

general, the number of sectors must be changed along with the number of sectors used by each FAT. The FAT uses 1.5 bytes for each sector. Also each directory entry uses 32 bytes. The sixty four entry directory is normally enough for any processor memory disk drive.

SUMMARY

The memory disk device driver presents a basic block device which can be implemented on any system which runs DOS V2.x or later. It implements the basic block device functions but not the IOCTL operations. The memory disk can be easily changed by modifying the parameters in the BPB or modifying the support code with extended memory boards.

Replacing the memory disk specific code with tested code for a new disk drive would be required to develop a device driver for the new disk drive. These routines can be tested before being placed into a device driver for easier debugging. The major modification would be placing any initialization code after the end of the resident device driver code. This allows the space used by the initialization procedures to be returned to the operating system after the driver is initialized.

The next article will present a character device driver. It uses the same front end as the memory disk device driver. However, the rest of the device specific code is different. (M)

Bill Wong is the President of Logic Fusion, Inc., 1333 Moon Drive, Yardley, PA 19067, a systems software development firm.

ONE COMPUTER & TWO PRINTERS?



\$65.95

Plus \$3.25 Shipping

30 Day Money Back Guarantee

Stop Plugging & Unplugging!
Use your choice of TWO printers from ONE parallel cable with the simple flick of a switch.

One Parallel Plug In... Two Parallel Plugs Out!
Connect any Standard Centronics Parallel Cable directly to the switch. Two 3 1/2' cables (included) attach to each printer.

Simply Call 1-800-TO-ASK-US! 8am-5pm PST
(CA 415/567-4067)

To Order: Write or call Tipz Direct, Inc. Visa/MC or Money Orders please. Shipping within 48 hrs. via UPS.

Shopping? Mail \$1.00 for our Catalog of Unique PC Accessories

Tipz

P.O. Box 690,
San Francisco, CA 94101-0690 MSJ-210

Source for MDISK.ASM

```

; === Loadable Memory Disk Device Driver ===
;
; by: William G. Wong
; Logic Fusion, Inc.
; 1333 Moon Drive
; Yardley, PA 19067
;
; =====
;
; V1.00 02-12-86 Initial version
;
;
; === Constant Definitions ===
;
; --- Request Header Structure Definitions ---
rh_length      equ 0          ; (byte) length field
rh_unit        equ 1 + rh_length ; (byte) unit field
rh_command     equ 1 + rh_unit  ; (byte) command code
rh_status      equ 1 + rh_command ; (word) status field
rh_reserved    equ 2 + rh_status ; (8 bytes)

rh_size        equ 8 + rh_reserved ; request header size

; --- Request header status values ---
s_done         equ 0100h      ; done, no errors
s_busy         equ 0200h      ; busy, no errors

e_protect      equ 8000h      ; error: write protect
e_unknown unit equ 8001h      ; error: unknown unit
e_not_ready    equ 8002h      ; error: not ready
e_command      equ 8003h      ; error: unknown command
e_crc          equ 8004h      ; error: bad CRC
e_bad_length   equ 8005h      ; error: bad structure length
e_seek         equ 8006h      ; error: bad seek
e_media        equ 8007h      ; error: unknown media
e_not_found    equ 8008h      ; error: sector not found
e_paper        equ 8009h      ; error: out of paper
e_write        equ 800ah      ; error: write fault
e_read         equ 800bh      ; error: read fault
e_general      equ 800ch      ; error: general error not listed above

; --- Non-destructive read parameter block ---
rh_read_data   equ rh_size ; (byte) non-destructive data

; --- Input/output parameter block ---
rh_media       equ 0 + rh_size ; (byte) media descriptor
rh_buf_offset  equ 1 + rh_media ; (word) transfer buffer offset
rh_buf_segment equ 2 + rh_buf_offset ; (word) transfer buffer segment
rh_buf_size    equ 2 + rh_buf_segment ; (word) transfer buffer size
rh_start       equ 2 + rh_buf_size ; (word) transfer starting sector

m_fixed        equ 0f8h      ; media: fixed disk
m_ss9          equ 0fch      ; media: single sided, 8 sectors/track
m_ds9          equ 0fdh
m_ss8          equ 0feh
m_ds8          equ 0ffh

; --- Build BPB parameter block ---
rh_bpb         equ 1 + rh_media ; preceded by media descriptor
rh_tbl_offset  equ 4 + rh_bpb ; (dword) bpb buffer address
rh_tbl_segment equ 2 + rh_tbl_offset ; (dword) bpb table offset/segment

; --- Media Check parameter block ---
rh_check       equ 1 + rh_media ; preceded by media descriptor
; (byte) media check result

mc_changed     equ -1         ; media has changed
mc_maybe       equ 0          ; media may have changed
mc_same        equ 1          ; media has not changed

; --- Initialize parameter block ---
rh_units       equ 0 + rh_size ; (byte) number of units supported
rh_end_offset  equ 1 + rh_units ; (word) end address of driver
rh_end_segment equ 2 + rh_end_offset
rh_bpb_offset  equ 2 + rh_end_segment ; (word) BPB array address
rh_bpb_segment equ 2 + rh_bpb_offset

```


; --- Device Driver Header Attribute Definitions ---

```

a_input      equ    0001h      ; standard input device
a_output     equ    0002h      ; standard output device
a_nul        equ    0004h      ; NUL device
a_clock      equ    0008h      ; CLOCK$ device
a_dos        equ    0       ; DOS block device ( bit 13)
a_not_dos    equ    2000h     ; non-DOS block device ( bit 13)
a_ioctl      equ    4000h     ; IOCTL functions supported
a_block      equ    0       ; block device (bit 15)
a_character  equ    8000h     ; character device (bit 15)

```

; === Device Driver Header Definition ===

```
cseg segment para public 'CODE'
```

```
driver proc far
```

```
assume cs:cseg,ds:cseg,es:cseg
```

```

dd      -1          ; last driver in chain
dw      a_block + a_dos ; driver attribute
dw      dev_strategy ; offset to strategy routine
dw      dev_interrupt ; offset to interrupt routine
db      1           ; number of devices or device name
db      7 dup ( ? ) ; filler for block device

```

; === Device Driver Tables ===

; --- BIOS Parameter Block Table and Entries ---

```
bpb_table dw bpb ; one entry for each unit
```

; --- Boot record for initialization ---

```

boot_record db 3 dup ( 0 ) ; non-bootable (no jump instruction)
             db 'DOS 2.1' ; identification

```

```
boot_bpb_offset equ ( $ - bootrecord )
```

```

bpb:
bytes_in_sector dw 512 ; BIOS Parameter Block
                 db 1 ; bytes/sector
                 db 1 ; sectors/cluster
bpb_reserved dw 1 ; reserved sectors
bpb_fats db 2 ; number of FAT's
bpb_root dw 64 ; directory entries in root
bpb_total dw 360 ; total number of sectors
bpb_media db m ss8 ; media type byte
bpb_fat_size dw 2 ; sectors/FAT

```

```
bpb_size equ ( $ - bpb )
```

```

dw 1 ; sectors/track
dw 1 ; number of heads
dw 0 ; hidden sectors

```

```
boot_size equ ( $ - boot_record )
```

; --- Request Header Address set by dev_strategy ---

```

rh_address dd 1 dup ( ? ) ; request header base address
rh_offset equ rh_address
rh_segment equ rh_address + 2

```

; --- Request Header Command Dispatch Table ---

```

cmd_table dw initialize ; initialize driver
           dw media_check ; media check
           dw build_bpb ; build BPB
           dw ioctl_read ; IOCTL read
           dw read ; normal read
           dw check_input ; non-destructive read/status
           dw input_status ; input status
           dw input_flush ; flush input buffers
           dw write ; normal write
           dw write_verify ; normal write with read verify
           dw output_status ; output status
           dw output_flush ; flush output buffers
           dw ioctl_write ; IOCTL write

```

; === Common Device Driver Routines ===

; --- Device Driver Strategy Routine ---

```
; es:bx == request header address
```

```

dev_strategy proc far
               mov cs:rh_offset,bx
               mov cs:rh_segment,es
               ret
dev_strategy endp

```

; --- Device Driver Interrupt Routine ---

```

dev_interrupt proc far
               push ax ; save registers used
               push bx
               push cx
               push dx
               push di
               push si
               push ds
               push es
               cld ; clear direction flag
               push cs ; setup small memory model
               pop ds ; ds := program segment
               les bx,rh_address ; es:bx := request header index
               mov si,es:rh_command[bx] ; si := request command (byte)
               and si,0ffh ; si := request command
               add si,si ; si := word table offset
               call word ptr cmd_table[si] ; ax := command result
               lds bx,cs:rh_offset ; ds:bx := request header index
               mov rh_stats[bx],ax ; update request status
               pop es ; restore registers
               pop ds
               pop si
               pop di
               pop dx
               pop cx
               pop bx
               pop ax
               ret

```

```
dev_interrupt endp
```

```
; **** END OF DEVICE INDEPENDENT PORTION OF DRIVER ****
```

; === Memory Disk Working Variables ===

```
mdisk_segment dw 0 ; memory disk segment set by initialize
```


Up To Your Ears In Alligators?

If that sounds familiar, you need Write-Hand-Man™, the multi-function pop-up desktop organizer that works neatly with existing software for CP/M™ 2.2 and 3.0 systems. Write-Hand-Man eliminates that swamped feeling with tools that will get you organized. Write-Hand-Man comes with a 4-function, floating-point, 14 digit Calculator – Notepad – Two-week Appointment Book, File and Directory viewing – Phonebook with dialing – Cut and Paste – Key Redefinition – ASC II table. Even add your own applications.



Clear the swamp from your desktop.
Order Write-Hand-Man today. \$49.95

CA residents add 6.5% tax. Sorry, no credit cards or purchase orders.
Specify: 8" or which 5" format
CP/M 2.2 or 3.0 format

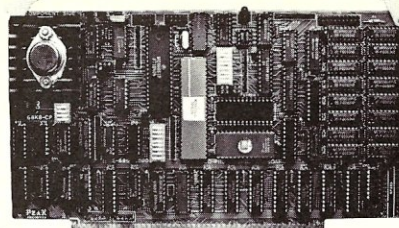
30 day guarantee

™ Write-Hand-Man – Poor Person Software

™ CP/M – Digital Research

Poor Person Software

Dept. 204
3721 Starr King Circle
Palo Alto, CA 94306
(415) 493-3735



68K8-CP

Expand Your System with a 68000 CoProcessor

Peak Electronics' 68K8-CP is a high performance 68000 software development package designed to easily integrate into your existing S-100 system. The package consists of the 68K8-CP coprocessor card, CP/M-68K, and a software toolkit that includes a UNIX V7 compatible floating point C compiler and a symbolic debugger.

Any system running CP/M®-2.2, CP/M-3.0 or CP/M-86 can be running CP/M-68K within minutes without any change in existing hardware or software. This card does not replace your current processor. All of the original system's devices (RAM, disks, and other peripherals) are immediately available to the user of CP/M-68K. All files can be accessed by whichever operating system is currently active. Control is transferred between operating systems with a simple one line command.

Features:

- Does not replace your current CPU card or software
- Includes CP/M-68K with UNIX® V7 compatible floating point C compiler and a symbolic debugger
- All developed C and Assembly code is fully relocatable and ROMable
- 8 or 10Mhz CPU with no wait state RAM
- 128K bytes of RAM expandable to 512K
- 2 serial and 1 parallel I/O ports
- IEEE-696-1983, S-100 Compatible
- 30 day money back guarantee
- 1 year parts and labor warranty

Complete Package: \$995.00
VISA or Master Card Accepted



P.O. Box 700112, San Jose, CA 95170-0112
(408) 253-5108



Micro/Systems Journal Software

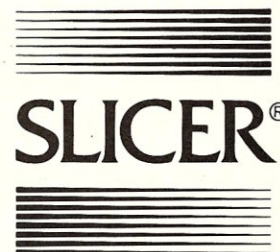
Software appearing in Micro/Systems Journal is released into the public domain via the SIG/M and PC/BUE Public Domain Software Libraries. It may thus be copied at many computer club meetings and downloaded from many RBBS bulletin board systems.

The Software may also be obtained directly from us. We currently have two disks of software available:

- Disk #1 - Software from 1985 issues
- Disk #2 - Software from 1986 issues

\$12/disk (includes shipping and handling). Outside North America add \$5 for shipping (via air). Ordering both disks deduct 10%. Send check (drawn on U.S. bank) or postal money order to: Micro/Systems Journal, Box 1192, Mountainside NJ 07092. Specify disk format.

THE PROVEN MICRO FAMILY for all the INDUSTRY APPLICATIONS



*Specials

New Low Prices!

*Multi-User Slicer Combo	\$950.00
Slicer SBC A&T (No RAM)	
Slicer 1 MEG Board A&T (Complete)	
CCP/M by Digital Research Inc.	
Slicer SBC A&T 256K	\$445.00
*Slicer SBC A&T 128K	\$350.00
with double deck sockets for additional 128K	
*Slicer SBC A&T (No RAM)	\$295.00
*Slicer 1 MEG Board A&T	\$450.00
*Slicer 1 MEG Board (Full Kit)	\$295.00
Slicer PC Expansion Board A&T	\$495.00
CCP/M (Digital Research Inc.)	\$250.00
MS DOS (Micro-Soft Corp.)	\$175.00
*New Slicer Bios for MS DOS	Call

Other kit forms available - Enclosures & other support hardware in stock - Call or write for latest information & prices!

SLICER COMPUTERS INC.

2543 Marshall Street N.E.
Minneapolis, MN 55418

(612) 788-9481
BBS (612) 788-5909


```

; ===== Memory Disk Device Driver Code =====
;
; ----- Driver support functions (near functions) -----
;
; es:bx == request header
; ds == cs
; All other registers are usable.
;
; ax := result status
;
; ----- Initialize driver -----

```

```

initialize proc near ; initialize driver

    lea ax,mdisk_data ; ax := end of driver
    mov cl,4 ; cl := paragraph size log2
    shr ax,cl ; ax := paragraphs in driver
    mov dx,cs ; dx := driver segment
    add ax,dx ; ax := memory disk segment
    mov mdisk_segment,ax ; update for subsequent transfers
    lea si,boot_record ; ds:si := boot sector for disk
    xor di,di
    mov es,ax ; es:di := boot sector on disk
    mov cx,boot_size ; cx := size of boot sector data
    rep movsb ; initialize boot sector

    mov ax,cs:bpreserved ; ax := first FAT sector
    mov bx,cs:bpreserved ; bx := number of sectors
    call destination_address ; es:di := first FAT, cx := size
    push di ; save first FAT address
    push es

    mov al,cs:bpreserved ; al := media byte
    mov [di],al ; update FAT media byte
    mov word ptr 1[di],0ffffh ; allocate initial sectors
    add di,3 ; adjust FAT index
    sub cx,3 ; adjust FAT size
    xor al,al ; al := 0
    rep stosb ; clear rest of FAT

    mov bx,cs:bpreserved ; bx := sectors/FAT
    mov ax,bx ; ax := sectors/FAT
    add ax,cs:bpreserved ; ax := second FAT sector
    call destination_address ; es:di := second FAT, cx := size
    pop ds
    pop si
    rep movsb ; ds:si := first FAT index
    ; copy first FAT to second FAT

    mov ax,cs:bpreserved ; ax := number of directory entries
    mov cl,5 ; cl := size of entry log2 (32 bytes)
    shl ax,cl ; ax := directory size
    mov cx,ax ; cx := directory size
    xor al,al ; al := 0
    rep stosb ; zero directory

    mov ax,cs:bpreserved ; ax := total number of sectors
    mov bx,1 ; bx := filler to prevent overflow
    call source_address ; ds:si := end of driver/disk
    mov ax,ds ; ax := end of driver segment
    lds bx,cs:rh_address ; ds:bx := request header address
    mov byte ptr rh_units[bx],1 ; return number of units
    mov rh_end_offset[bx],si ; return ending address of driver
    mov word ptr rh_bpb_offset[bx],offset bpb_table ; return bpb table
    mov rh_bpb_segment[bx],cs ; return BPB table address

    push cs
    pop ds
    mov dx,offset initok
    mov ah,9
    int 21h

```

```

mov ax,s_done ; ax := done, no errors
ret

```

```

initok db 'MDISK initialized',10,13,'$'

```

```

initialize endp

```

```

; ----- Media check -----
;
; Memory disk is non-removeable media

```

```

media_check proc near ; media check
    mov es:byte ptr rh_check[bx],mc same
    mov ax,s_done ; ax := function done, no errors
    ret

```

```

media_check endp

```

```

; ----- Build BPB -----
;
; Read boot sector and copy into BPB buffer.
; Adjust request header values.

```

```

build_bpb proc near ; build BPB
    mov ax,0 ; ax := sector to read
    mov bx,1 ; bx := number of sectors
    push cs ; save driver segment
    call source_address ; ds:si := address, cx := bytes to move
    pop es ; es := driver segment
    add si,boot_bpb_offset ; ds:si := sector offset
    mov di,offset bpb ; es:di := BPB buffer address
    mov cx,bpb_size ; cx := BPB size in bytes
    rep movsb ; copy BPB from boot sector to buffer

    mov al,cs:bpreserved ; al := media byte
    lds bx,cs:rh_address ; ds:bx := request header address
    mov byte ptr rh_media[bx],al
    mov word ptr rh_tbl_offset[bx],offset bpb
    mov word ptr rh_tbl_segment[bx],cs
    mov ax,s_done ; ax := function done, no errors
    ret

```

```

build_bpb endp

```

```

; ----- Read from device -----

```

```

read proc near ; normal read
    push es
    pop ds ; ds := request header segment
    push rh_buf_offset[bx] ; save destination offset
    push rh_buf_segment[bx] ; save destination segment
    mov ax,rh_start[bx] ; ax := first sector to read
    mov bx,rh_buf_size[bx] ; bx := number of sectors
    call source_address ; ds:si := source sector, cx := size
    pop es
    pop di ; es:di := buffer address
    rep movsb ; copy sector to buffer
    mov ax,s_done ; ax := transfer done
    ret

```

```

read endp

```



```

; --- Write to device ---
write proc near
    push es
    pop ds
    push rh buf offset[bx]
    push rh buf segment[bx]
    mov ax, rh_start[bx]
    mov bx, rh_buf_size[bx]
    call destination_address
    pop ds
    pop si
    push cx
    rep movsb
    pop cx
    mov ax, s_done
    ret
; normal write
; ds := request header segment
; save source offset
; save source segment
; ax := first sector to read
; bx := number of sectors
; es:di := destination, cx := size
; ds:si := buffer address
; save for possible verify
; copy sector to buffer
; cx := transfer size
; ax := transfer done

```

write endp

; --- Write and verify ---

```

write_verify proc near
    call write
    std
    dec si
    dec di
    repe cmpsb
    jnz verify_error
    mov ax, s_done
    ret
; normal write with read verify
; cx := size, es:di/ds:si at end
; reverse direction
; index last byte of source
; index last byte of destination
; repeat until end or difference
; skip if transfer failed
; ax := transfer done, no errors

```

```

verify_error:
    mov ax, e_write
    ret
; ax := write error

```

write_verify endp

; === Support Functions ===

; --- Unimplemented functions ---

```

unimplemented proc near
    mov ax, e_command
    ret

```

unimplemented endp

```

ioctl_read equ unimplemented ; IOCTL read
check_input equ unimplemented ; non-destructive read/status
input_status equ unimplemented ; input status
input_flush equ unimplemented ; flush input buffers
output_status equ unimplemented ; output status
output_flush equ unimplemented ; flush output buffers
ioctl_write equ unimplemented ; IOCTL write

```

; --- Compute Memory Disk Sector Address ---

```

;
; ax = sector (base 0)
; bx = number of sectors to move
;
; cx := bytes to move
; ds:si := sector address

```

```

source_address proc near
    mov dx, cs:bytes_in_sector
    mov cl, 4
    shr dx, cl
    mul dx
    add ax, cs:mdisk_segment
    mov ds, ax
    mov ax, cs:bytes_in_sector
    mul bx
    mov cx, ax
    or cx, cx
    jnz sector_done
    dec cx
    mov si, si
    ret
; dx := bytes/sector
; cl := paragraph size log2
; ax := paragraphs/sector
; ax := sector paragraph offset
; ax := sector segment
; ds := sector segment
; ax := bytes/sector
; ax := bytes to move
; cx := bytes to move
; zero := full segment move
; skip if valid size (non-zero)
; cx := maximum transfer size (0ffffh)
; si := 0 (byte offset)

```

source_address endp

; --- Compute Memory Disk Sector Address for Destination ---

```

;
; ax = sector (base 0)
; bx = number of sectors to move
;
; cx := bytes to move
; es:di := sector address

```

```

destination_address proc near
    call source_address
    mov di, si
    push ds
    pop es
    ret
; es:di := sector address

```

destination_address endp

; === Memory Disk Data Area ===

```

; Align to paragraph boundary for easy computation of the sector address
if (( $ - driver ) mod 16 )
    org ( $ - driver ) + ( 16 - (( $ - driver ) mod 16 ) )
endif
mdisk_data equ $ ; memory disk starts here

```

; === End of Memory Disk Device Driver ===

```

driver endp
cseg ends
end

```


Magnum Digital PRO-180 & FD-100

by Steven D. Kapplin

For the past year the 8-bit world has been abuzz with excitement over Hitachi's announcement of its new 8-bit CPU, the HD64180. This new chip provides many new features, yet is completely compatible with Intel's Z-80 processor. It provides extended memory addressing (up to 512k), on board DMA channels and serial ports, a memory management unit (MMU), and several new instructions, including hardware 8-bit multiply. The new chip operates at 6 Mhz (a 9Mhz version is also available) and promises greater throughput than a Z-80 running at the same clock speed.

Byte magazine (September and October 1985), ran a two-part series, by Steve Ciarcia describing a single board computer project based on the HD64180, dubbed the SB-180. And Micro/Systems Journal (May/June 1986) carried a construction article on building a 64180 S-100 CPU card. Now Magnum Digital, Inc. of Stover, Missouri is marketing the PRO-100, an S-100 CPU card, using the HD64180, and the FD-100, a companion floppy disk controller.

THE PRO-180

The PRO-180 features the Hitachi HD64180 micro-processor running at 6 Mhz. The board includes sockets which can accommodate up to 64K of RAM or ROM memory. The sockets can be configured with up to 32K of ROM in one socket (Hitachi modules, 2716s, etc. can be used) and 32K of RAM in the second socket. Thus, the PRO-180 can be configured as a stand-alone dedicated processor complete with ROM and RAM. Our evaluation board was supplied with two Hitachi RAM modules providing 64K of static RAM. The RAM can be assigned to any 64K bank using a dip switch.

The PRO-180 has three serial ports and a bus-port for remote control of the CPU. Two serial ports operate at speeds up to 38.4K baud. The third port is a clocked serial port which can operate at speeds up to 300K baud for use with local area networks. The CPU has plenty of jumpered options making it easy to configure the board to the user's system.

THE FD-100 DISK CONTROLLER

The floppy disk controller is designed around the Western Digital 2793 chip. Four

An S-100 CPU System Using The Hitachi HD64180

drives can be connected in any combination of 5¼" and/or 8". The 2793 is capable of auto-density and side select, so mixing different drives presents no problem for this disk controller. The card has a socket provided for a ROM, and ours was supplied with Magnum Digital's monitor ROM. A dip switch is provided for setting the port boundary for the 2793, which requires 8 consecutive ports. Few support chips are required for the FD-100. The board is very sparsely populated.

GETTING THINGS UP AND RUNNING

Magnum Digital supplies standard CP/M and Echelon's Z-System as available operating systems. At the time we were reviewing the PRO-180 the Z-System was not available (it is now available.) The standard CP/M supplied included Magnum Digital's customized versions of SYSGEN and MOVCPM as well as two specialized programs, SETSYS and CONTRL. SETSYS is used to alter the disk parameter blocks and enable the OS to read foreign disk formats. Magnum Digital currently supports IBM 3740 and System 34 disk formats as well as its own native formats for 8" and 5¼" disks. Magnum Digital's native formats include SSDD and DSDD 8" disks configured with 1,024 byte sectors, 9 sectors per track. This provides 675K and 1.35 meg of formatted storage for SSDD and DSDD, respectively. Magnum Digital also supports 48 and 96 tpi 5¼" drives in DSDD format, only.

CONTRL is a program enabling one to read individual sectors from a disk of virtually any format. Sectors are read into memory where they can be dumped or written to another disk. This provides a rudimentary disk editing capability.

I installed the PRO-180 and FD-100 in an old IMSAI 12-slotter (no front panel). Magnum Digital supplied two CP/Ms, one

on 5¼" disk and one on 8". Both were double-sided double-density. Unfortunately, my 8" drives are single-sided, so I started out using the 5¼" disk. The system boots up in the ROM monitor first. The monitor provides two disk boot options: boot from a 5¼" in drive A or boot from an 8" in drive A. The 5¼" system booted up with a short sign-on message announcing that this was a 60K CP/M system. Magnum Digital's implementation of CP/M is fairly straight forward. The five inch system was configured with drives A and B as DSDD 5¼", 48 tpi. Drives C and D were configured as DSDD 8". Installation was relatively simple and I had no difficulty getting the system to boot and run.

The PRO-180 monitor is not extensive, but provides all the essential requirements of a monitor including commands to display, move, and alter memory. Disk boot from the monitor is provided, and CONTRL permits reaccess to the monitor from the OS.

ADVANTAGES OF THE PRO-180/FD-100

After spending good money for this hardware, what can you expect in the way of performance? I benchmarked the PRO-180 against my 4 Mhz CCS Z-80 system. Using the Sieve and Dhamptone benchmarks, I found that the PRO-180 is exactly twice as fast as my Z-80 for processor-bound activities. Total throughput was tested using the Dhamptone benchmark which includes disk I/O operations. This benchmark ran 35 percent faster on the PRO-180. I tested other disk-bound operations as well. The PRO-180 loaded Wordstar 16 percent faster than my Z-80. Magnum's BIOS for the PRO-180 uses multiple sector reads and writes which provides for very quick disk routines. A scroll from the top to the bottom of a 32K text file took 15 seconds on a 5¼" disk using the PRO-180 and FD-100 controller. The same scroll required 31 seconds on my CCS Z-80 and controller.

There is little question that the PRO-180 and FD-100 provide a significant increase in system speed and throughput. The 6 Mhz HD64180 is twice as fast as a 4 Mhz Z-80 due to the faster clock speed and the fact that the HD64180 instructions are

HIGH PERFORMANCE S-100

THE MOST POWERFUL 8 BIT MICROPROCESSOR YOU CAN BUY FULLY Z80
CAPATIBLE, CP/M COMPATIBLE, Z-SYSTEM COMPATIBLE

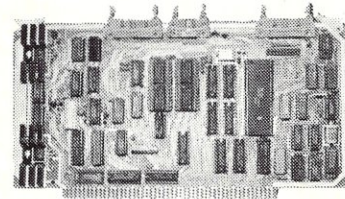
*Use the PRO-180 processor with your current controller, or buy the Magnum MDC-1 at
great savings and have one of the most powerful microcomputers anywhere.*

PRO-180 PROCESSOR BOARD

(HD64B180 for S100's)

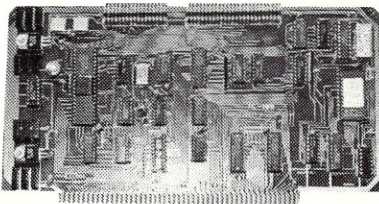
The PRO-180, using Hitachi's new high integration HD64B180 processor, packs a lot of performance in a small area, while maintaining the modularity of the S100 structure.

- Executes all Z80 instructions. The HD64180 instruction set is a superset of the Z80, so the Pro-180 will run all software written to run on the Z80 or 8080 processors.
- 6 MHZ or greater operation.
- Hardware multiply (8 bit multiply, 16 bit result).
- Two standard built-in serial I/O ports that require no wait states. Standard baud rates up to 38,400 baud.
- Clocked serial port for multiprocessor/multicomputer communications at up to 300,000 baud on a pair of wires.
- Has 65,000 I/O ports (not just 256 like older processors).
- HD64180 has 19 address lines for directly addressing 512K bytes of memory with built-in MMU.



- Two channel DMA controller that can directly address 512K bytes of memory.
- The PRO-180 implements the full 24 bit extended address bus allowing it to address up to 16 megabytes of memory.
- Versatile interrupt controller with 8 internal and 4 external sources.
- Uses "off the shelf" components. No "hard to get" PALS or Gate Array devices.
- Built-in, programmable wait state generator. This allows wait states to be inserted only when they are needed for specific devices, allowing faster modules to run at full speed.
- Two channel, 16 bit, programmable, reload timer. The timer can be used for many timer functions, including real time clock applications.

MDC-1 Floppy Disk Controller



- The MDC-1 floppy disk controller can control any mixture of up to four 5 1/4 or 8" drives.
- Onboard shadow ROM.
- I/O mapped. Does not take up any memory space.
- Available with bios for CP/M 2.2, or with complete Z-SYSTEM.
- Runs 48tpi drives, 96tpi drives, or 48tpi diskettes in 96tpi drives.
- Runs with 8 bit or 16 bit processors.

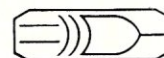
PRO-180 processor w/64K static RAM	\$349
w/32k static RAM and ROM monitor	\$349
without RAM or ROM	\$299

MDC-1 floppy disk controller w/bios	\$199
w/complete Z-SYSTEM & shadow ROM	\$279

COMPLETE SET:

PRO-180 w/64k SRAM, MDC-1, Z-SYSTEM, S-ROM, UTILITIES	\$569
--	--------------

VISA, M/C, C.O.D. Orders - Missouri residents add sales tax.



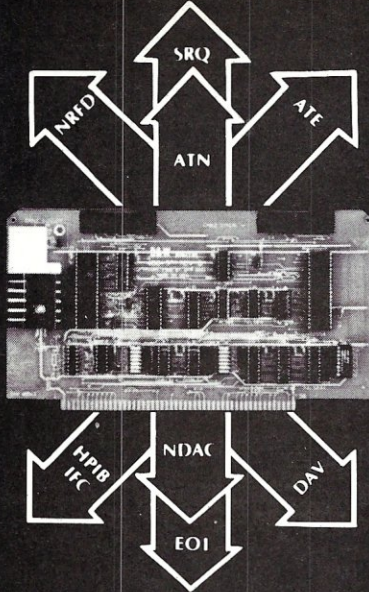
**magnum
digital, inc.**

Route 2, Box 223, Stover, MO 65078

Mail Orders to: Magnum Digital Inc.
3211 South Providence Rd., Suite 302
Columbia, Missouri 65203
(314) 449-7199

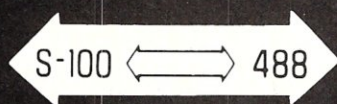
or call: For technical assistant (314) 377-4419

Trademarks: CP/M2.2 by Digital Research, Z SYSTEMS by Echelon, Inc., Z80 by Zilog, Inc., 8080 by Intel Corp.



THE 488+3

IEEE 488 TO S-100
INTERFACE



- Controls IEEE 488 (HP1B) Instruments with an S-100 computer
- Acts as controller or device
- Basic and assembly language drivers supplied
- Meets IEEE 696 specification
- Industrial quality burned in and tested up to 125K bytes/sec under software control 3 parallel ports (8255-5)
- \$375



D&W DIGITAL, INC.
20655 Hathaway Avenue
Hayward, California 94541
(415) 887-5711

mostly 3 clock cycles rather than the Z-80's 4 clock cycles.

PROBLEMS AND DISADVANTAGES

One of the major problems we discovered is that the PRO-180 will be difficult to get working with another disk controller. We were not able to get the PRO-180 to work by itself with either a CCS 2422 disk controller or with the Morrow Disk Jockey DMA controller. Magnum Digital informed us that a boot ROM would more than likely have to be prepared to permit the PRO-180 to work properly with other disk controllers. This means that installing a PRO-180 without the FD-100 would not be a trivial task. I believe that only experienced OEMs or systems houses would be able to provide a working system. If you buy both the PRO-180 and FD-100, however, installation should prove relatively simple. In its current configuration, however, the PRO-180 can only boot from the monitor. No provisions are made to permit a cold boot directly from disk, bypassing the monitor boot routines. This is a minor, but annoying inconvenience.

A second disadvantage is that the PRO-180 has no parallel port for driving a printer. Furthermore, Magnum Digital uses the HD64180's serial port 1 for the console. Port 1 has handshaking implemented, but Port 2 does not. Therefore, no hardware handshaking is available for the serial printer port (Port 2). It would have been a cleaner implementation to have the console on Port 2, as most terminals require no handshaking. Thus, Port 1 would provide hardware handshaking for serial printers. Under the current configuration you would have to implement handshaking in software for a serial printer. If you have a parallel printer, then you will need to get a serial interface for it! This is not a serious problem, but would represent an added cost to the conversion from a Z-80 system to the PRO-180.

A third problem exists for those of you with front panels. The PRO-180 does not support front panels. It does, however, provide a jumper to enable/disable MWRT for those whose front panels generate MWRT.

DOCUMENTATION

The PRO-180 comes with a thick manual, but most of it is the technical data taken from the Hitachi HD64180 technical manual. The actual board documentation is somewhat sparse. If you are not an experienced hacker or have friends who are, I would not tackle configuring the PRO-180 and FD-100. If you are attempting to install only the PRO-180, be sure you have expert help available. The manuals are definitely written with the OEMer in mind. The casual user would find the manuals difficult to understand. A complete source listing of the monitor ROM is provided in

the documentation. There is a Table of Contents, but no index. The source for the monitor and BIOS are provided on disk to permit user customization.

The manual for the FD-100 is not as overtly attractive as that for the PRO-180, but I found it to be a few degrees more useful. Again, however, this manual is not intended for the novice.

Unfortunately, no documentation is provided for CP/M nor for the customized programs. SYSGEN, SETSYS, and CONTRL are all menu-driven, and thus easily used, but they have many features which require explanation. I understand that documentation will be available in the near future. The current documentation provides no assistance for system regeneration. This would be expected of a product aimed more for the OEM or system house market than toward the end-user.

FINAL OBSERVATIONS

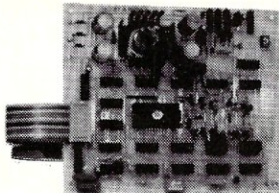
The PRO-180 and FD-100 make an excellent upgrade of a Z-80 or 8080 system. I was impressed by the speed increase. The PRO-180 will certainly be enhanced by the addition of Echelon's Z-System which will help take advantage of the new Hitachi chip. The PRO-180 should be purchased with the FD-100, unless you are looking forward to a long and difficult task of ROM burning and customization. In fact, I would not recommend the PRO-180 to the casual end-user. It is definitely better to have an OEM or system house configure a system for you, if you are not an experienced hardware user. Were it not for many friends in the area, I would have found installing the PRO-180 by itself an impossibility. The addition of the FD-100 controller made the installation easier, but not readily adaptable to my system.

The PRO-180 and FD-100 appear to be a reasonably good hardware buy, but needs more extensive and better documentation for less experienced users. The need for a special boot ROM will make use of the PRO-180 without the accompanying disk controller extremely difficult for anyone other than OEMs or system houses. Although not confirmed, there may be some problem in adapting the PRO-180 to systems using the Morrow Disk Jockey controller. The PRO-180 provides the advantage of being able to move up to the HD64180 processor without scrapping one's S-100 hardware. The price is quite attractive: the two board set is currently being offered for \$569.00 complete with CP/M (and, now, probably Echelon's Z-System.) (μ)

Steve Kapplin is Associate Professor of Real Estate and Finance at the University of South Florida. He has been an active "hacker" since 1981. He can be reached via CompuServe ID 70055,1021 or through the Finance Department at the University.

CP/M, MS-DOS EPROM PROGRAMMING SYSTEM

2708
2758
2516
2716
2732
2764



2732A
2764A
27128
27128A
27256
27CXX

- SOFTWARE AVAILABLE FOR CP/M-80, CP/M-86 AND MS-DOS SYSTEMS
- STAND ALONE BOARD - ELECTRONIC SWITCHING OF EPROM TYPES
- USES 24 VOLT XPMR FOR POWER - ALL SUPPLIES/TIMING ON BOARD
- NO PERSONALITY MODULES TO BUY - LARGE COMPREHENSIVE MANUAL

** PARALLEL PRINTER INTERFACE **

CONNECTS TO ANY PARALLEL PRINTER INTERFACE - USES 8 OUTPUT DATA BITS AND ONE INPUT DATA BIT (BUSY LINE). BUSY LINE IS A HIGH SPEED SERIAL INPUT. FULL EPROM READING AND PROGRAMMING UNIT MAY ALSO BE CONNECTED TO ONE 8 BIT INPUT/OUTPUT PORT.

** CONTROL PROGRAM COMMANDS **

- PROGRAM EPROM(S) FROM DISK
- READ DISK FILE INTO RAM
- READ EPROM(S) INTO RAM
- VERIFY EPROM IS BRASED
- DISPLAY/MODIFY RAM - (MONITOR MODE) WITH 11 SUB COMMANDS
- FILL-DUMP-XPER-EXAMINE-MODIFY-BIAS-PROGRAM-VERIFY, ETC.)
- SAVE EPROM(S) TO DISK
- PROGRAM EPROM(S) FROM RAM
- COMPARE EPROM WITH RAM
- COPY EPROM

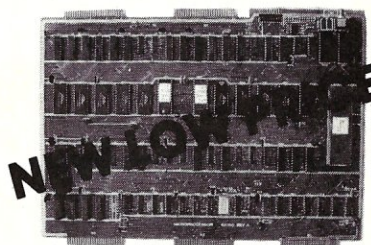
ASSEMBLED & TESTED UNIT WITH COMPLETE DOCUMENTATION AND SOFTWARE ON DISKETTE PARTS KIT WITH SOFTWARE AND DOC - \$179/PCB, SOFTWARE & DOC - \$69 SOFTWARE ON 8 & 5 1/4 DISK FOR IBM, OSBORNE, KAYPRO & OTHER PMTS

TO ORDER SEND CHECK, MONEY ORDER, WRITE OR CALL

ANDRATECH
P.O. BOX 222
MILFORD, OHIO 45150
(513) 752-7218

CALL OR WRITE FOR MORE INFORMATION -- ADD \$4.00 FOR SHIPPING OHIO RES. ADD 5.5% TAX -- VISA/MC. ACCEPTED -- \$3.00 FOR COD

M68000 SINGLE BOARD COMPUTER



On board 6-10 MHz CPU, 20K RAM, 32K EPROM, two RS-232, 16-bit port, 5-counter/timers expandable via Memory/FDC Board.

M68K CPU (bare board)	\$ 89.95
M68K CPU A&T (6MHz)	\$495.00
MD512K Memory/FDC (bare board)	\$ 89.95
MD512K Memory/FDC (128K)	\$495.00
FDC/Hard Disk interface option	\$150.00
M68KE Enclosure w/power supply	\$249.00
M68K Monitor EPROM's	\$ 95.00
M68K Macro Cross Assembler	\$195.00
4XFORTH OS w/assembler, editor	\$295.00
CP/M 68K OS w/"C" compiler	\$395.00

EMS Educational Microcomputer Systems

P.O. Box 16115
Irvine, CA 92713
(714) 854-8545

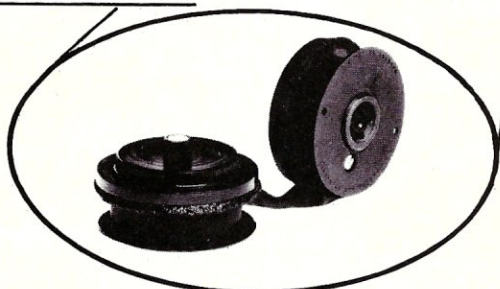
Announcing a New Product...

OPEN SPOOL PRINTER RIBBON THAT RE-INKS ITSELF

FOR USERS OF

**Okidata and Other
Open Spool Printers
Including Teletype**

MONEY BACK GUARANTEE



OUR SELF-INKING DEVICE WILL LAST 15 TIMES THE RIBBON YOU ARE NOW USING.

- 15 TIMES LESS RIBBON CHANGES
- NO MORE MESSY HANDS

CONTROLLED PRINTOUT DEVICES, INC.

P.O. BOX 869, BALDWIN ROAD
ARDEN, NC 28704

(704) 684-9044 • TELEX: (FILMON-AREN) 577454

CONTACT US BY MAIL, PHONE OR TELEX AND WE WILL FORWARD YOU A BROCHURE



POOR MAN'S NETWORK

Now you can implement networking on your own CP/M computers and share resources whenever you want. Each user can access files and printers on the other computer directly, without expensive hardware or switches, and without a communications program. Share floppies, RAM-disk, hard disks, and printers between two users. Works with most standard CP/M programs, like Wordstar, PIP, dBase II, etc.

Poor Man's Network works best with bidirectional parallel ports, or standard serial ports at 9600 baud or greater. Present version supports two computers only, and requires CP/M 2.2 on each computer. Uses only 6K of memory. Does not require a hard disk; runs on 8080, 8085, Z-80, HD-64180.

Poor Man's Network comes ready to run on BigBoard I and II, Xerox 820, NorthStar Horizon, NorthStar Advantage, Televideo TPC I, Kaypro (not 2000), Apple II (Super Serial required), S-100 with Compupro Interfacer 4. Can be installed on other computers by altering one of the assembler overlay files provided. Each disk contains drivers for all the specified computers.

Best of all
is the price: only **\$69!**

Specify disk format: 8" SSD, NorthStar, Kaypro, or Apple CP/M. Phone orders accepted; sorry, no CODs or Purchase Orders. Payment may be by certified cheque, money order, Visa, or Mastercard. Personal cheques from outside Canada require up to 6 weeks to clear. Canadian orders please pay in Canadian dollars; others in US dollars. Price includes diskette, manual and postage. Ontario residents please add provincial sales tax.

Note: If you use another operating system, such as CP/M 3 or PC-DOS, send us your name and address, but no money, and we will notify you when it is available for your system.

ANDERSON TECHNO-PRODUCTS INC.

947 Richmond Road, Dept 5
Ottawa, Ontario K2B 6R1, Canada

Telephone **613-722-0690** for more information or to order.

CP/M is a trademark of Digital Research; Z-80 is a trademark of Zilog; Wordstar is a trademark of MicroPro; dBase II is a trademark of Ashton-Tate. Poor Man's Network is a trademark of Doug Anderson Techno-Products Inc.

Declare & Define C Variables In One File

by Bill Rogers

The statements for defining and declaring an external variable in C are different¹. Further, only the definition statement can include initialization. A combined definition and declaration scheme can lead to ease in maintenance and to increased program clarity.

By means of the C preprocessor, it is possible to make a single statement define, declare and initialize an external variable.

The storage-class specifier *extern* is used to declare an external variable. To define an external variable, no storage-class specifier is used.

The external variable is both defined (no storage-class specifier) and declared ("extern") with the pseudo-storage-class specifier "EXTERNAL". See FLAG_1 and FLAG_2 in the header file "flags.h" in

Figure 1. Further, if the defined variable is initialized, the initial value is conditionally compiled if "EXTRINIT" is currently defined. See FLAG_2 in the header file "flags.h" in Figure 1.

To define the external variables FLAG_1 and FLAG_2, use the sequence given in the c source file "definer.c" in Figure 2. "EXTERNAL" is replaced with nothing. Since "EXTRINIT" is defined, the initialization clause for FLAG_2 is included. The resulting C preprocessor output is given in Figure 4.

To declare the external variables FLAG_1 and FLAG_2, use the sequence given in the c source file "declarer.c" in Figure 3. "EXTERNAL" is replaced with "extern". Since "EXTRINIT" is not defined, the initialization clause for FLAG_2 is not included. The resulting C pre-

processor output is given in Figure 5.

The origin of this scheme is the section on BDS C program conversion in the Computer Innovations C User's Manual².

[μ]

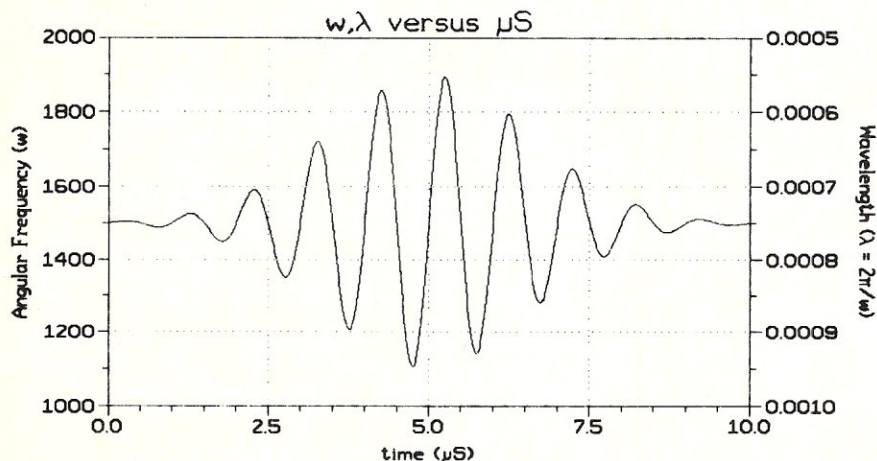
¹ "A declaration announces the properties of a variable (its type, size, etc.); a definition also causes storage to be allocated." Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language. Prentice-Hall, Englewood Cliffs, NJ. 1978. pg 76.

² Computer Innovations, Inc., Optimizing C86 User's Manual. Document Version 2.3. May 1985. pg 1-32.

Bill Rogers is a programmer with Concentric Associates, Secaucus NJ.

GRAPHICS

starting at \$49⁹⁵



SCI-GRAF: Produces line, scatter, and high/low plots on Epson or IBM compatible dot-matrix graphics printers. Supports images up to 1680x1712 dots (over 3 million pixels!), log & linear scales, graphs in 7 colors (on a JX-80), batch processing. Requires MS-DOS 2 or 3, 256k ...\$99⁹⁵

FONTEdit: Easy-to-use full screen font editor allows you to create and modify fonts for our SCI-GRAF program. Create Greek, math, and custom symbols in 3 sizes. Supports SCI-GRAF's proportional spacing. Requires IBM compatible 320x200 pixel on-screen graphics & IBM compatible keyboard\$39⁹⁵

GRAF 3.0: Produces bar, pie, line, scatter, and high/low plots on Epson, IBM, C.Itoh, or NEC compatible dot-matrix graphics printers. Supports 14 fill-in patterns, 8 point-plotting symbols, automatic scaling, labeling, and legend creation. MS-DOS 2 or 3, 192k\$69⁹⁵ CP/M-80: Requires 64k (54k TPA) \$49⁹⁵

MSC

Microcomputer
Systems
Consultants

27 Forest Avenue Port Jefferson Station New York 11776-1820 (516) 928-7493

```
/* flags.h */

EXTERNAL int FLAG_1;
EXTERNAL int FLAG_2
#ifdef EXTRINIT
    = 0
#endif
;
```

Figure 1. Header File Defining/Declaring External Variables.

```
/* definer.c */

#define EXTERNAL
#define EXTRINIT
#include <flags.h>
#undef EXTRINIT
#undef EXTERNAL
```

Figure 2. Program File Defining External Variables.

```
/* declarer.c */

#define EXTERNAL extern
#include <flags.h>
#undef EXTERNAL
```

Figure 3. Program File Declaring External Variables.

```
definer.c 1:
definer.c 2:
flags.h 1:
flags.h 2:
flags.h 3: int FLAG_1;
flags.h 4: int FLAG_2
flags.h 6:          = 0
flags.h 7:
flags.h 8:          ;
```

Figure 4. Preprocessor Output from "definer.c".

```
declarer.c 1:
declarer.c 2:
declarer.c 3:
flags.h 1:
flags.h 2:
flags.h 3:extern int FLAG_1;
flags.h 4:extern int FLAG_2
flags.h 7:
flags.h 8:          ;
```

Figure 5. Preprocessor Output from "declarer.c".

Can't find files?

Contains the most advanced features:

- automatically reads 65 character file comment lines
- 45 character disk title
- ASCII BOBFILE containing file comment lines, file names and sizes, and disk title
- handles binary or text files
- directories and sub-directories
- floppy or hard disks

PC - DOS 2.0 & 3.0; CP/M 2.2 & 3.0; TURBODOS

US Residents	\$49.95 US funds
Canadian Residents	\$49.95 Canadian funds
other countries	\$54.95 US funds



plus \$3.00 postage and handling

Ontario Residents add 7% PST

Extra DOS or sites \$15.00 each to maximum of \$75.00

**B
O
B
C
A
T**

R&L MicroServices Inc, Box 15955, Stn F, Ottawa, Ontario, K2C 3S8 (613)225-7904

No source code for
your REL files?

REL/MAC

converts a REL file in the Microsoft™ M80 format to an 8080 or ZILOG™ Z80 source code MAC file with insertion of all public and external symbols.

- REL/MAC makes MAC source files
- REL/MOD lists library modules
- REL/VUE displays the bit stream
- REL/PAK includes all of the above
- 8080 REL/MAC demo disk \$10.00

REL/PAK for 8080 only \$99.95

REL/PAK for Z80 & 8080 \$134.95

on 8"SSSD disk for CP/M™ 2.2

Send check, VISA, MC or C.O.D. to



MICROSMITH
COMPUTER TECHNOLOGY

P.O. BOX 1473, ELKHART, IN 46515

1-800-622-4070

(Illinois only 1-800-942-7317)

CACHE22 + CP/M 2.2 = CP/M Max!

CACHE22 is a front-end system program that buries all of CP/M 2.2 in banked memory. It helps 8080/Z80 computers to survive by providing up to 63.25K of TPA, plus the ability to speed disk operations, eliminate system tracks, and run Sidekick-style software without loss of transient program space. Complete source and installation manual, \$50.00.

CP/M is a trademark of Digital Research Inc.
Sidekick is a trademark of Borland International



MIKEN OPTICAL COMPANY

53 Abbett Avenue, Morristown, NJ 07960
(201) 267-1210

The UNIX File

by Ian F. Darwin

This column discusses the UNIX operating system. If you have comments or questions about UNIX or this column, please write to Ian Darwin at Box 603, Station F, Toronto, Ontario, Canada M4Y 2L8. If you have access to the uucp network, mail "ihnp4!darwin!ian". I can't always answer immediately, but I will get back to you; electronic mail gets answered first!

A friend asked me to check whether a title he was planning to use for a UNIX book had already been taken. I called into Dialog Information Services and looked into the Online edition of Books In Print (OBIP). I didn't find the title he wanted, but I did find no less than seventy-one books on or about UNIX. Included in the group are some books I've never heard of, catalogs of UNIX products, and some duplicate entries. Several books, that I know of, are not listed. My breakdown of the 71 is as follows: 45 textbooks, 8 reference cards, 7 books on word processing and typesetting, 7 product catalogs, and a few books on languages, applications, and other topics. Rather than try to review each of them (maybe I should write a book about all the other UNIX books!), I thought I'd describe one or two good or interesting books from each area.

INTRODUCTORY TEXTS

There are so many introductory texts now that it's hard to know what to say. The "best" by far is *Introducing the UNIX System* by McGilton and Morgan. Another good book is *A UNIX Primer* by Lomuto and Lomuto. *UNIX For People* by Brown et al. is a good non-technical introduction despite some factual errors. Mark Sobell's *A Practical Guide to the UNIX System* is a little more detailed; there's also a version of this book customised for System V. Good introductions for programmers or those who've used other computer systems include Kaare Christian's *The UNIX Operating System* and Steve Bourne's *The UNIX System*.

WORD PROCESSING & TYPESETTING

Morris Krieger's *Word Processing on the UNIX System* does a credible job of

A UNIX Book Roundup

explaining how to use *nroff/troff* and the editor. Krieger's presentation uses the *-ms* formatter macros. A shorter book on the same topic by Constance Brown, Jack L. Falk, and Richard D. Sperline is *Preparing Documents with UNIX* (Prentice-Hall, 1986, ISBN 0-13-699976-X). At about two hundred pages (and half the size of Krieger's book), this book also uses *-ms*, and it presents templates for letters, resumes, and other common documents rather than detailing the principles of *nroff/troff*. The book includes a glossary of text formatting and typesetting terms. Nahrain Gehani's *Document Formatting & Typesetting on the UNIX System* provides similar coverage using the *-mm* macros, at a level between the Brown and Krieger books, but also suffers from some problems in typesetting. Several other recent works cover editing, or editing and formatting, but I haven't seen copies of them yet.

PROGRAMMING TEXTS

The UNIX Programming Environment by Brian Kernighan and Rob Pike is highly recommended; it covers the gamut of topics on programming for UNIX. Marc Rochkind's *Advanced UNIX Programming* concentrates on UNIX system calls as used by the C programmer, with numerous examples including the source for a simple (non-programmable) UNIX shell.

I have just received a copy of *Introduction to Compiler Construction with UNIX* and haven't read it carefully yet; it includes coverage of *yacc* and *lex* as well as introducing compiler-writing techniques.

BOOKS ON UNIX OPERATING SYSTEMS

There is no widely-available text on how current versions of UNIX work. A commentary by John Lions described the Sixth Edition in some detail, but was only

available from AT&T; they have apparently stopped distributing it. Doug Comer's *Operating Systems: The XINU Approach* presents the design and source code for an educational tool; XINU is an OS that is in some ways like a subset of UNIX. The code is presented in the book, and is available from the publisher. Ric Holt's *Concurrent Euclid, the UNIX System and Tunis* gives some good descriptions of how major parts of UNIX work; Tunis is an experimental UNIX work-alike system written at the University of Toronto.

SYSTEM ADMINISTRATION?

I don't know of any good books on UNIX system administration. Online Books In Print lists *DOS UNIX: Becoming a Super User*; this sounds like it might be about system administration, but it could also be a book about IBM PC's. *UNIX System Security* by Wood and Kochan covers a range of topics on security, and looks useful despite a few factual errors about what features are in what version of UNIX.

UNIX CATALOGS

The grandest of the UNIX catalogs is the */usr/group UNIX Products Directory*. The 1986 edition is the size of a telephone book, although it's not printed in the same type style. There are two or three listings per page, each describing some hardware or software offering related to UNIX. It pretty much covers the whole market.

OTHERS

Finally a UNIX book in another language; Silvester's *Guide argent du system UNIX* (Springer-Verlag, 1985) is the first foreign-language book on UNIX listed in OBIP. But it appears to be a translation, possibly of Silvester's *The UNIX System Book*.

Most interesting title: *Tricks of the UNIX Masters* anonymous, Sams Publishers. Sounds like it could be "Secrets of the Jedi Masters", but I haven't seen a copy yet.

There are many books that I haven't mentioned; most are books that I haven't

continued on page 93

LSEARCH 3.0

dBASE III

— A GENERAL-PURPOSE CATALOGUE SYSTEM WITH FAST, MULTIKEY BOOLEAN SEARCH —

- User abstracts **keywords** from source materials:

REAL ESTATE LISTINGS
LEGAL TRANSCRIPTS AND DEPOSITIONS
COMPANY AND CLIENT PROFILES
PERSONNEL PROFILES

RESUME'S
PATIENT CHARTS
ARTICLES, BOOKS, CLIPPINGS
WRITER'S OR RESEARCHER'S NOTES

HOLIDAY TOURS AND TRIPS

- Stores the abstracts in a simple universal file structure.
 - Searches the "library" for relevant abstracts using **general multikey boolean forms** composed of **lists of keywords**.
 - List syntax for boolean forms is simple; lists are stored to file for easy editing.
 - **Search algorithm is FAST.**
 - **Every application handled in the same way.**
-
- The all-purpose file structure has just two fields: abstract id number, and keyword. One record for every keyword. User sets the widths of the fields.
 - **Categorization of keywords** is accomplished by a variable one-character prefix on the keyword (any printable ASCII character).
 - Search lists come in four types: **MATCH**, **NONMATCH**, **GRTR/=** and **LESS/=**.
 - Multilist queries come in two types: **ALL** and **ANY**.
 - An item satisfies an **ALL** query if it satisfies all the lists (**AND** logic).
 - An item satisfies an **ANY** query if it satisfies any list (**OR** logic).
 - A list may have any number of keywords, and any mix of categories.
 - **Search output is a list of "hits"**, the id numbers of the abstracts which satisfy the query, stored to a file.

Resume Example		
Categories are: T = title, F = function, G = year graduated, P = product area, \$ = salary, D = degree, M = major, S = school		
Sample Abstract #43	Sample Queries	
id keyword 43 T_Asst. Director 43 F_Sales 43 G_73 43 P_Fabrics 43 P_Yarns 43 P_Clothing 43 \$_37000 43 D_MBA 43 M_Marketing 43 S_Columbia U.	(a) experience in Sales and Marketing; not a V.P. or Director; 10-15 years work experience; current salary not higher than \$50K. ALL MATCH F_Sales MATCH F_Marketing NONMATCH T_Director T_V.P. GRTR/= G_71 LESS/= G_76 LESS/= \$_50000	(b) not a steel industry executive (Director, V.P. or C.E.O.) ANY NONMATCH T_Director T_V.P. T_C.E.O. NONMATCH P_Steel

- Searching is fast because all entries across all categories are indexed simultaneously — all "search keys" are inverted at once.
- The system is menu-driven except for file creation and editing.
- Users should know dBASE III, but don't need to be programmers.
- dBASE III source code is included, so if you are a programmer you can adapt the system to taste.
- The system provides an option for automatic encoding of keywords and phrases into 3 or fewer bytes, leading to dramatic disk savings. (An application of this type is included with every order: abstracts of dBASE III material from TechNotes.)

To order, please send a check for \$49 + \$3.50 (S&H) to

Thinker's Apprentice / 392 Central Park West, Apt. 12X / New York, NY 10025
(212) 222-5050

(Residents of New York State, please add state and local sales tax.)

dBASE III is a registered trademark, and TechNotes is a copyright publication of Ashton-Tate.

Hochstrasser's Modula-2

by Edward J. Joyce

Modula-2, like most computer languages, traces its roots to the main-frame world. In 1979, Niklaus Wirth and a team of software engineers designed the first compiler on a DEC PDP-11 computer at the Swiss Federal Institute of Technology (Eidgenössische Technische Hochschule, or ETH) in Zurich. From there the language made its way down the computer totem pole, starting with the Lilith (a Modula-2 computer engine built by Wirth and Richard Ohran) and spreading to the IBM PC, Macintosh and Apple II. Meanwhile, no version surfaced for CP/M-80 and it seemed as if an inexplicable wall had arisen between Wirth's latest brainchild and one of the most popular operating systems.

That wall came down in July of 1985 when Peter Hochstrasser announced the first Modula-2 implementation for CP/M-80. Distributed by a company that bears his name, Hochstrasser's Modula-2 brings a state of the art language to the legions of loyal users who still compute in the eight-bit CP/M world.

Hochstrasser started on CP/M Modula-2 while at ETH in Switzerland. He and three others wrote the software as part of an academic thesis. During the course of the project, they consulted with the primary architects of the language, including the maestro of Modula-2 himself, Niklaus Wirth. Indeed, the acknowledgements section of the user's manual reads like a *Who's Who* in Modula-2.

The fact that Hochstrasser's product originated at ETH, where Wirth concocted both Pascal and Modula-2, lends authenticity to the package. The software is not a subset or stripped-down version. On the contrary, it even includes floating point arithmetic, assembly language interfaces, extensive library modules and other items one might not expect from an initial CP/M implementation.

Strictly speaking, the product should be called Modula-2 for Z80 CP/M, which is how the vendor refers to it. The compiler, linker and related programs execute only on a Z80 microprocessor and the code generated, which can be embedded in ROM, contains Z80 instructions. The

8080 and 8085 microprocessors, generally included under the umbrella of CP/M compatibility, are not supported.

The system generates Z80 machine code. In other words, the final output of the linker is an executable object file (or .COM file in CP/M parlance). The object file contains all run-time support routines needed. It does not rely on auxiliary files for input and output. For example, with the popular Sieve of Eratosthenes benchmark program, the compiler generates a 2K object file that is ready for execution.

The executable object file approach stands in contrast to pseudocode, also called P-code or M-code, produced by some Pascal and Modula-2 compilers. These systems typically compile programs faster, but they lose their advantage during execution because an interpreter must interpret the pseudocode. The machine code generated by Hochstrasser's four-pass compiler, on the other hand, is executed directly by the Z80 and the difference shows in superior performance.

Since Modula-2 is by design a simple language, much of its power stems from library modules supplied with the implementation. Hochstrasser provides the standard module library defined by Wirth, namely, elementary procedures for input and output from a terminal or file, math functions and storage management. This basic offering is then augmented by a robust library of utility modules.

MathLib0, the standard math library, goes by the title MathLib, without a suffix, in this system. Besides the eight math functions prescribed by Wirth, it contains a procedure for exponentiation. According to the user's manual, MathLib corresponds with the math library proposed by MODUS (Modula-2 Users Association).

One of the strongest assets of the utility library rests in a module called OpSys. OpSys opens the door to the nitty-gritty of the CP/M operating system. It gives the programmer direct access to CP/M's BDOS and BIOS. For example, selecting a disk or checking the keyboard for a character is only a procedure call away through OpSys.

Programmers writing business applications will be pleased with the library's string manipulation modules. These modules handle command line parsing, massaging of text data, and conversions between the internal format of real and integer numbers and their displayable ASCII forms. What programmer hasn't had to build one of these routines from scratch during his or her career?

The library's file management routines take the grunt-work out of retrieving and recording data on disk by providing sequential and random file access mechanisms. Although vanilla Modula-2 makes no provision for random access, Hochstrasser includes a module that closely resembles FileSystem. FileSystem is the file management module Wirth constructed for his Lilith computer. It supports file positioning at the byte and record level with error checking.

Another library component which further enhances the attractiveness of this system as a general purpose tool for software development lies in the chaining module. It allows one program to invoke or *chain* to another program. The module accommodates sharing of data between the chained programs, too. The chain facility is invaluable for multi-program applications such as a mailing list printing, where the first program selects mailing addresses according to specified criteria, a second program sorts the addresses, and a third program prints the addresses on mailing labels.

Although the developers of this system profess to be avowed proponents of Modula-2, they did not turn their backs on other programming languages. They recognize those situations where a programmer needs to incorporate software written in other dialects into a Modula-2 program. Hochstrasser's system offers an assembler interface for meshing Modula-2 with foreign tongues. The user's manual spells out the interface, describing register usage, parameter passing, machine level representation of data, and other details.

Hochstrasser's linker combines the assembler routines with Modula-2 code. The format of the relocatable code files

Modula-2 for Z80 CP/M

Manufacturer

Hochstrasser Computing AG
Chratzstrasse 14
8954 Geroldswil
Switzerland

U.S. Distributors

The Alternate Source
704 North Pennsylvania Ave.
Lansing, MI 48906
(517) 482-8270
(800)253-3200, X-700 (outside Michigan)
(800)632-7818, X-700 (in Michigan)

Workman & Associates

112 Marion Ave.
Pasadena, CA 91106
(818) 796-4401

System Requirements

CP/M Operating System
Z80 microprocessor
56K RAM
Two disk drives

Price

\$165

processed by the linker bears a close resemblance to Microsoft relocatable (.REL) files. An appendix in the manual outlines the linker's format. For easy conversion of Microsoft .REL files to Hochstrasser's format, a utility program is delivered on the distribution disk.

Hochstrasser thoughtfully included Modula-2 source code for the library modules on the software's three distribution disks. Consequently, the curious programmer can trace the complete flow of logic when a call is issued to a library routine. Availability of source code is a refreshing change from other high level language products that hide input and output operations in *black box* interfaces or run-time packages which inevitably complicate debugging.

For example in Hochstrasser's Modula-2, reading a character from a sequential disk file through the procedure Read of standard module InOut generates the following events. The Read procedure invokes (or *imports* in Modula-2 terminology) Texts, a module which processes streams of text from a file or the console. Texts brings in a module called SeqIO for sequential input and output from files. Finally, SeqIO issues the appropriate calls to the operating system through CP/M's BDOS to retrieve the desired sector of data. This last step will be bypassed, if SeqIO already has the desired character in a buffer obtained from a previous read operation.

The layered approach allows the programmer to choose any of several levels

for handling input and output ranging from high-level, device independent access to low-level, step-the-disk-drive type dialogue.

Owners of budget CP/M systems, like me, whose disk domain consists of a mere dual set of single density floppy drives might shy away from a software package that spans 533K of space on three distribution disks. Although the software comes delivered that way, you can operate with less than three disks online. I was able to configure a working system on two disks by judiciously manipulating the library modules and deleting source code files, which are not required for compilation and linking.

Lately, some software entrepreneurs have the nerve to sell language compilers without cross reference generators under the guise of "professional programming tools." In my mind, a compiler without a cross reference is about as useful as a disk without a directory. Fortunately, Hochstrasser didn't take the shortcut of leaving out this essential element—a stand-alone program generates the cross reference in a disk file or directly on the printer.

Other software supplements include several source code examples which serve as introductions to CP/M Modula-2. There's the ubiquitous Sieve of Eratosthenes program and a solution to the eight Queens problem in chess. For recreational relief between intense periods of Modula-2 programming, Hochstrasser delivers a game program called Take, in which the user and computer alternately remove sticks from a pile in increments of one, two or three until the loser is left with the last stick (the program plays a mean game).

The qualities inherent in the software, comprehensiveness and correctness, carry over to the user's manual, too. The 250-page typeset document thoroughly explains the compiler, linker, library modules and other system components. Pascal keyboard-jockeys will especially value the language introduction which spells out the differences between Modula-2 and Pascal.

Overall, Hochstrasser has done a commendable job of introducing Modula-2 to the CP/M-80 arena. Programmers dealing with both applications and systems software will find a well-rounded tool in this package. The \$165 price covers royalty fees, which accounts for one more reason why Hochstrasser's Modula-2 warrants serious attention among CP/M software jocks. [M]

Ed Joyce, author of Modula-2: A Seafarer's Manual and Shipyard Guide (Addison-Wesley, Reading, Mass., 1985), is an ardent fan of Swiss cheese, Swiss chocolate, Swiss army knives and Swiss computer languages.

DeSmet C

— now with —

LARGE MEMORY OPTION

C88 Compiler \$109

Full K&R + V7 extensions
inline asm
assembler, linker, librarian
full screen editor (SEETM),
8087 & S/W floating point

D88 Debugger \$50

Set Breakpoints by line number
or function name
Examine Global/Local variables
by name
Show C source while debugging
Both D88 and User displays

Large Memory \$50 Model Option

32-Bit Pointers (Full Megabyte
Addressing)
Fast Access of Static Data
Works with D88

— plus —

DOS Link \$35
convert to .OBJ files

Graphics (+src) \$35

Hacker \$25
source for start-up,
RAM.COM & more

Make \$50
full UNIX-level

Tools (+source) \$35

XARRAY \$39
Large Arrays (+source)

C WARE CORPORATION

505 W. OLIVE, SUITE 767
SUNNYVALE, CA 94086 USA
(408) 720-9696 TELEX: 358185
We accept
VISA, MC & American Express

More Loadable BIOS Drivers For CP/M

by Ted Carnevale

Cal Sondgeroth's article on loadable drivers for CP/M 2.2, (Micro/Systems Journal vol.1 #2 pp.66-71, 1985) was a real eye-opener for me. His XBIOS and XLOAD programs showed a convenient, straightforward way to intercept calls to the BIOS.

It would be useful to have utilities that would tell what BIOS extension modules had been loaded, and allow removal of extensions as necessary without rebooting the system. Furthermore, since some installations of CP/M have additional entries in the BIOS jump table, it would be nice if the BIOS extension loader XLOAD could install modules independently of such nonstandard enhancements.

One way to do this is to add a header to the BIOS extension module. This header would contain data needed by XLOAD and other utilities that manipulate BIOS extension modules, and could be checked by utilities that need to know if a BIOS extension was installed. I revised XLOAD and XBIOS accordingly, and wrote two other programs in C that would describe installed BIOS extensions and allow me to remove them.

Intercepting BIOS Calls

Details of Sondgeroth's approach are best obtained from his article (especially Fig.2), but a short description is appropriate here. The BIOS extension module XBIOS contains a special warm boot routine and two jump tables: one that is a copy of the prior contents of the BIOS's own jump table, and another that contains jump instructions to whatever BIOS service routines the user prefers. It is this second table that is the key to the utility of the BIOS extension module.

XLOAD moves XBIOS into place at the top of the TPA, copies the original BIOS's jump table into the proper location in the XBIOS module, and modifies the original table so that the BIOS jumps to the XBIOS. It also changes locations 6 and 7 to point to the start of the XBIOS module, thereby reducing the TPA and protecting the XBIOS from being overwritten. One of the nice features of XLOAD is that it shows how the bit map at the end of an

An Enhancement To Tell The Loaded BIOS Extension & Add & Remove Extensions Without Reboot

.SPR file can be used to relocate code. If necessary, this could be modified to accommodate .PRL files instead.

Several BIOS extension modules can be loaded, one after another. The jump tables in the demonstration listing of Sondgeroth's article are constructed so that jumps to BIOS routines (except for warm and cold boot) will pass through all of the BIOS extension modules, finally reaching the BIOS itself. Obviously, a practical BIOS extension module will contain one or more jumps to its own service routines.

The programs that I wrote carry the examples of XLOAD and XBIOS a step further. I wanted to be able to identify what modules had been loaded, and in what order. Also I wanted to be able to remove them one at a time.

To make this easier, I rearranged XBIOS and put a header at its beginning that contains certain useful items (see NUXB in Listing 1). The header is exactly 32 bytes long. After the module is loaded, its first three bytes will be "jmp bdos," the same as in the original version of XBIOS.

The fourth and fifth bytes allow the design of generalized utilities to handle BIOS extension modules. The fourth byte

is used by the BIOS extension loader (LXB—Listing 2) and other utilities to find where the jump tables begin. Its value equals the total length of the *version* and *function* strings (see below), which must be 255 or fewer bytes.

The fifth byte in the header specifies how many jump instructions are to be copied from the BIOS into the BIOS extension. This is handy for use with *enhanced* BIOS's that have jump tables with additional functions.

Bytes 6-16 are presently unused. Bytes 17-32 contain the string "xBIOS installed\$" which serves two purposes. When a BIOS extension has been installed, this message is shown on the console at every warm boot. It is also checked by utilities in order to verify that a BIOS extension is in place.

Immediately after the fixed length header comes a data area that contains *version* and *function* strings that tell the date and revision number of the BIOS extension, and what it does. These strings are terminated by nulls so that they can be readily manipulated by C programs.

The copy of the previous contents of the BIOS jump table comes next (OLDTBL), followed by the BIOS extension's own jump table. There didn't seem to be any reason to preserve the cold boot jump in this revision of the BIOS extension module.

The XLOAD program has been changed to LXB (Listing 2), which accommodates the revised structure of the BIOS extension module. The major change was to include code to find the OLDTBL area of the BIOS extension. Like XLOAD, LXB will load and relocate a BIOS extension module from a file with an .SPR extension. SPR files can be produced from REL files with Digital Research's LINK by using the "OS" option switch.

NUXB and LXB were not designed to intercept BDOS calls. A flexible way to do this might be to use the first two reserved bytes in the BIOS extension header as storage for the BDOS entry address, while the first three bytes in the header would be a jump to a BDOS intercept routine in the extension module itself. If BDOS calls are

to be simply passed on to the BDOS, the intercept routine could consist of the instructions LHL D RESERV ! PCHL. Otherwise, the value in register C would be tested and the appropriate action taken. If this approach to BDOS intercepts is used, the loader would have to be revised to patch the BDOS address into the appropriate location in the header.

Two utilities that are useful for dealing with these BIOS extension modules are CXB.C and RXB.C. CXB (Listing 3)

shows the starting address and the version and function strings of all BIOS extensions that have been installed, starting with the one lowest in memory (i.e. the one most recently loaded). RXB (Listing 4) removes the most recently installed BIOS extension module, leaving the rest in place. These were written for the Software Toolworks C80 compiler, but should run with little or no change with any other compiler.

Loadable BIOS extensions bring additional flexibility to CP/M 2.2. The revised BIOS extension module and loader, and the utilities CXB and RXB, make it easier than ever to modify CP/M's input/output facilities *on the run*. u

Ted Carnevale, an assistant professor of neurology at SUNY Stony Brook NY, is involved in research on information processing in neurons.

LISTING 1 - NUXB.ASM

```

TITLE 'Loadable bios module for CP/M v.2.2'
PAGE 62

;After C.Sondgeroth's XBIOS—see M/SJ vol.1 #2 p.66 et seq.
;7/12/85 Modified to allow use of general xbios loader,
;and enable use of special utility to remove xbios's--NIC
;
;Compile with RMAC, then link like so:
;
;   link nuxb[os]
;
;to produce .SPR file for use by bios extension loader LXB

CR      EQU      0DH
LF      EQU      0AH
CDISK   EQU      4      ;addr holds currently logged drive
PRINTF  EQU      9      ;bdos "print string" function

;These macros code for strings to identify bios extension.
;To allow easy manipulation by C programs, each macro
;should end with a null.
;NOTE: total length of data produced by these two macros
;must be 225 bytes or less.

VERSION MACRO
    ;; revision number & date
    ;; terminate this macro with a null
    DB 'v.1.1 7/12/85'
    DB 0
    ENDM

FUNCT   MACRO
    ;; tell what it's for
    ;; put in lines as desired, with cr lf as needed
    ;; restrictions:
    ;; --terminate this macro with a null
    ;; --total length of data produced by version & funct
    ;; macros must be 255 bytes or less
    DB 'Tests new version'
    DB 0
    ENDM

;*****
;This header identifies xbios module & provides vital info
;used by loader and other utilities

;"continuation jump" to bdos--patched to correct location
;by xbios loader. If this is the only xbios module, this
;jmp goes directly to the original bdos. Otherwise, it
;goes to the xbios module that was previously loaded.
BDOS:   JMP $-$

;distance from end of this header to the local copy
;of the original bios's entry table, i.e. total length
;of the version and function strings.
DB OLDTBL-HDREND

;distance from end of this header to the local copy
;of the original bios's entry table, i.e. total length
;of the version and function strings.
DB OLDTBL-HDREND

;how many jump instructions copied from the bios's table
DB (TBLEND - OLDTBL)/3

;Next 11 bytes reserved for future use
RESERV: DB 0
        DW 0,0,0,0,0

;Next string, which is exactly 16 bytes long and starts
;exactly 16 bytes from the first location in the xbios, is
;checked by utilities that manipulate resident xbios modules.
LABEL:  DB 'xbios installed$'

HDREND  EQU $

;end of fixed-length header
;*****

;strings that tell what it is & what it does
VERSION
FUNCT
;When bios extension is first installed, contents of
;original bios jump table are copied into following table.
;These jumps allow calls to any bios routines that are not
;supposed to be intercepted.
;If your bios has additional jumps that you might want to
;intercept, add more entries to this table

OLDTBL  EQU $

WBOOT   JMP $-$
CONST   JMP $-$
CONIN   JMP $-$
CONOUT  JMP $-$
LIST    JMP $-$
PUNCH   JMP $-$
READER  JMP $-$
HOME    JMP $-$
SELDSK  JMP $-$
SETTRK  JMP $-$
SETSEC  JMP $-$
SETDMA  JMP $-$
READ    JMP $-$
WRITE   JMP $-$
LISTS   JMP $-$
SECTRAN JMP $-$

TBLEND  EQU $

;This table should be same length as OLDTBL.
;It defines new entry points for bios routines that are to
;be modified. Any routines that are not to be intercepted
;should jump to corresponding entries in previous table.
;Eventually they will reach original bios entry points.

NEWTBL  EQU $

JMP XWBOOT ;to special bios extension "warm boot"
;routine. This is only jump that cannot go
;to corresponding entry in previous table.
;rest of these may be modified to intercept bios call
JMP CONST
JMP CONIN
JMP CONOUT
JMP LIST
JMP PUNCH
JMP READER
JMP HOME
JMP SELDSK
JMP SETTRK
JMP SETSEC
JMP SETDMA
JMP READ
JMP WRITE
JMP LISTS
JMP SECTRAN

;Special warm boot routine that intercepts CP/M's attempt
;to overwrite call to bios extension's own jump table. It
;bypasses reload of CCP and reinitialization of low memory
;jump vectors that are part of a normal CP/M 2.2 warm boot.
XWBOOT: LXI SP,80H ;reset stack pointer
        LXI B,80H ;set default dma address
        CALL SETDMA
        LXI H,BDOS
        SHLD 6

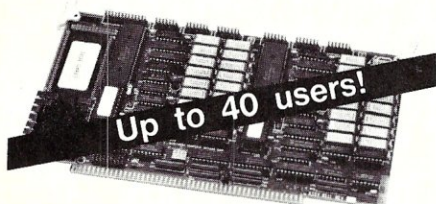
        ;fix "jmp bdos" at 0005 to point
        ;to this module.

        LXI D,CRLF
        MVI C,PRINTF
        CALL BDOS
        LXI D,LABEL
        MVI C,PRINTF
        CALL BDOS
        LHL 1
        LXI D,-1600H
        DAD D
        LDA CDISK
        MOV C,A
        PCHL ;jump into CCP

CRLF:   DB CR,LF,'$'
        END

```


Products With Expandability

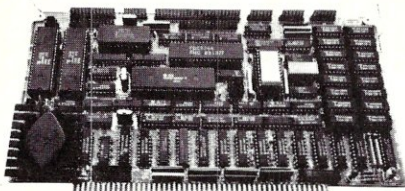


A two user Slave card based on Hitachi's Z80 compatible high speed, 10MHz super microprocessor.

Priced at **\$495^{00*}**

Features Include...

- 4-10 MHz Z80 Compatible HD64180
- 1/2 Megabyte Nonbanked Memory
- 2 Asynchronous Serial Ports To 38.4
- 1 High Speed Synchronous Port
- All Transfers Via 1.6 MHz DMA!!!
- Unique Expansion Port Offers;
 - 2 Additional Serial Ports or ...
 - 2 Parallel Ports or ...
- Real Time Clock With Battery Backup



The industry's fastest 8-bit Master CPU card with features superior to most 16-bit cards.

Priced at **\$495^{00*}**

Each Master Features...

- 4-10 MHz Z80 Compatible HD64180
- 1/2 Megabyte Nonbanked Memory
- 2 Asynchronous Serial Ports To 38.4
- 1 High Speed Synchronous Serial Port
- 4 Bi-directional Parallel Ports
- TurboDOS**, ZSYSTEMS**, CP/M**, & OASIS** Operating Systems
- FDC Simultaneously Controls 8", 5 1/4", & 3 1/2" Drives
- SASI/SCSI Interface
- Optional High Speed Hard Disk/File Access Tape Backup and True ETHERNET Controller

*Prices apply to 6 MHz, 64KB versions and are good for a limited time only on purchases of ten or more. For less than ten, please call.

**Trademarks: TurboDOS - Software 2000; ZSYSTEMS - Echelon; CP/M - Digital Research; OASIS - THEOS Software

ICD INTELLIGENT COMPUTER DESIGNS CORP.

23151 Verdugo Drive, Suite 113
Laguna Hills, CA 92653
(714) 581-7500

LISTING 2 - LXB.ASM

TITLE 'LXB---BIOS extension loader'
PAGE 62

;After C.Sondgeroth's XLOAD--see M/SJ vol.1 #2 p.66 et seq.
;Modified 7/12/85 by NTC for new xbios called nuxb

```
TRUE EQU OFFH
FALSE EQU NOT TRUE
INTRUP EQU FALSE ;TRUE if hardware uses interrupts routinely
HDLRNTH EQU 20H ;# of bytes in xbios's constant-length header
BOOT EQU 0 ;warm boot entry
BDOS EQU 5
FCB EQU 5CH ;default fcb
CONOUT EQU 2 ;console output func
OPEN EQU 15 ;bdos open file
READ EQU 20 ; read sequential
SETDMA EQU 26 ; set dma addr
CR EQU 0DH
LF EQU 0AH
```

ORG 100H

```
XLOAD EQU $
LXI H,0
DAD SP
SHLD STACK ;saves stack pointer
LXI SP,STACK ;& creates local stack
LDA FCB+1
CPI ' ' ;check for name of bios extension (xbios) file
JNZ XL20
CALL MSG
DB 'USAGE: lxb filnam',CR,LF
DB ' where filnam is a bios extension file',CR,LF
DB ' (.SPR extension assumed)'
DB 0
```

```
CPM$EXIT:
LHLD STACK
SPHL ;restore stack pointer
RET ;& bail out
XL20: LXI H,FCB+9 ;here starts file type
MVI M,'S' ;force .SPR extension
INX H
MVI M,'P'
INX H
MVI M,'R'
INX H
MVI C,24 ;# of bytes remaining in fcb
SLCLR: MVI M,0 ;clear the rest of the fcb to nulls
INX H
DCR C
JNZ SLCLR
LXI D,FCB
MVI C,OPEN
CALL BDOS ;open the xbios file
ANA A
JP SL50 ;if successful
CALL MSG ;else say failed
DB 'unable to find the bios extension file'
DB 0
```

```
JMP CPM$EXIT
;Read the bios extension file into low memory
SL50: LXI H,BUF ;hl = dma addr
SL60: PUSH H
XCHG ;de = dma addr
MVI C,SETDMA
CALL BDOS ;tell cpm where to put next sector
LXI D,FCB
MVI C,READ ;read next sector
CALL BDOS
POP H
LXI D,128
DAD D ;hl = next dma addr
ANA A
JZ SL60 ;not finished
;Relocate .SPR file's contents to high memory (just below
;ccp, or below last xbios module)
IF INTRUP
DI
```

```
ENDIF
LDA BOOT+2 ;calc page of ccp
SUI 16H
MOV H,A ;hl = ccp base page addr
LDA BDOS+2 ;current bdos page addr
CMP H
JNC REL1 ;if no prior xbios
LHLD BDOS+1 ;else get current top of tpa
MVI L,0 ;start on a page boundary
XCHG ;de = l+ top of tpa
LXI H,BUF+1 ;hl points to length of spr code
MOV C,M
INX H
MOV B,M ;bc = length of spr code
PUSH B
INR C ;round bc up to an integral # of pages
DCR C ;if c = 0, code ends on a page boundary
```



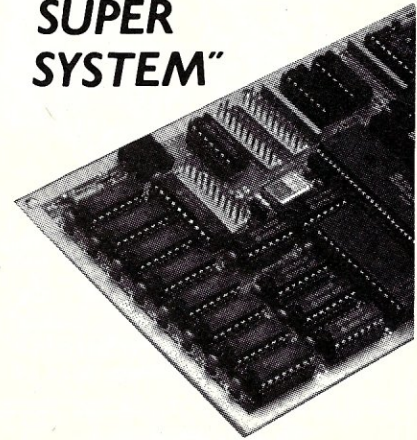
```

JZ REL2
INR B
REL2: MOV A,D      ;calc where to move the code, put result in de
      SUB B      ;a = bios page
      STA XBPAGE ;for future use
      MOV D,A
      MVI E,0    ;de = 1 + top of current tpa - code length
      LXI H,BUF+256 ;start of relocatable code
      POP B
      PUSH B     ;bc = actual length of spr code
      PUSH D     ;de = where to move it to (start of xbios)
      CALL MOVEM ;first move it up without relocation
      POP D      ;de = start of xbios
      LXI H,BUF+256 ;hl = start of relocatable code
      POP B      ;bc = length
      DAD B      ;hl = start of bit map
      PUSH D     ;save page offset
      XCHG      ;hl = addr of code, de = addr of bit map
      PUSH B     ;save byte count
;Get the next bit map byte and use it to direct
;relocation of 8 bytes
;Stop when end of relocatable code is found
RELNMAP: MVI C,8
         LDAX D
         MOV B,A ;b = map byte
         INX D   ;point to next map byte
;Shift & test relocation bits.
;If bit=0, don't alter code, else add offset for new base addr
RELNBYT: DCR C ;count bits in the byte
         JM RELNMAP ;get next byte
         MOV A,B
         RAL      ;get next bit
         MOV B,A
         JNC NOREL
         LDA XBPAGE ;offset
         ADD M      ;add offset to code
         MOV M,A
NOREL: INX H ;next location to consider
        XTHL ;hl = byte count
        DCX H
        MOV A,H
        ORA L ;done?
        XTHL ;hl = code pointer again
        JNZ RELNBYT ;no, keep working
;After done with relocation:
;1. fetch the original bios jump table and save a copy of it
;   in the new xbios module
;2. modify the original bios jump table so it points to the
;   corresponding locations in the xbios's jump table
        POP B ;remove counter from the stack
        POP H ;hl = start of new bios module
        PUSH H
;Next section is for new xbios--NTIC
;Copy original bios jmp table into xbios at OLDTBL
        INX H
        INX H
        INX H
;m = length of strings between end of header & start of jmp table
        PUSH H ;save place in the xbios header
        MOV A,M
        ADI HDRLNTH ;a = distance from start of xbios to
                    ;start of jmp table
        MOV L,A ;hl points to start of strings
        XCHG ;de is addr of xbios's copy of orig jmp
              ;table (dest)
              ;return to xbios preface for more
        POP H
        INX H
        MOV A,M ;# of jmp instr in table (omits jmp cold)
        STA JMPNUM ;for future use
        ADD M
        ADD M ;a=length of jmp table in bytes
              ;(omitting jmp cold)
        MOV C,A
        MVI B,0 ;bc = # of bytes to copy
        LHLD BOOT+1 ;hl=addr of 1st jmp to copy from bios table
        CALL MOVEM ;copy orig bios jmp table > oldtbl in xbios
        LHLD BOOT+1 ;hl = addr of jmp wboot in orig bios table
        LDA JMPNUM
        MOV C,A ;# of jmp instr to patch
;Now ready to patch original jmp table to point to xbios's own
;table.
NEWJMP: INX H ;patch old jmp table to point to
         MOV M,E ;corresponding entries in xbios
         INX H
         MOV M,D
         INX H
         INX D
         INX D
         INX D ;point to next entry addr in xbios jmp table
         DCR C
         JNZ NEWJMP ;to finish fixing the old jmp table
         POP D
         PUSH D ;de = addr of xbios
         LHLD BDOS+1 ;hl=destination-old bdos jmp (bdos entry)
         PUSH H ;saved on stack
         LXI H,BDOS+1 ;hl = where to patch jmp to xbios

```

Byte Magazine called it.

"CIARCIA'S SUPER SYSTEM"



The SB180 Computer/Controller

Featured on the cover of Byte, Sept. 1985,
the SB180 lets CP/M users upgrade to a
fast, 4" x 7 1/2" single board system.

- **6MHz 64180 CPU**
(Z80 instruction superset), 256K RAM,
8K Monitor ROM with device test, disk
format, read/write.
- **Mini/Micro Floppy Controller**
(1-4 drives, Single/Double Density,
1-2 sided, 40/77/80 track 3 1/4", 5 1/4"
and 8" drives).
- **Measures 4" x 7 1/2"** with mounting holes
- **One Centronics Printer Port**
- **Two RS232C Serial Ports**
(75-19,200 baud with console port
auto-baud rate select).
- **Power Supply Requirements**
+5V +/-5% @500 mA
+12V +/- 20% @40mA
- **ZCPR3 (CP/M 2.2/3 compatible)**
- **Multiple disk formats supported**
- **Menu-based system customization**

SB180-1

SB180 computer board w/256K
bytes RAM and ROM monitor
.....\$369.00

SB180-1-20

same as above w/ZCPR3, ZRDOS
and BIOS source.....\$499.00

-Quantity discounts available-

NEW

COMM180-M-S

optional peripheral board adds
1200 bps modem and SCSI
hard disk interface.

TO ORDER
CALL TOLL FREE
1-800-635-3355

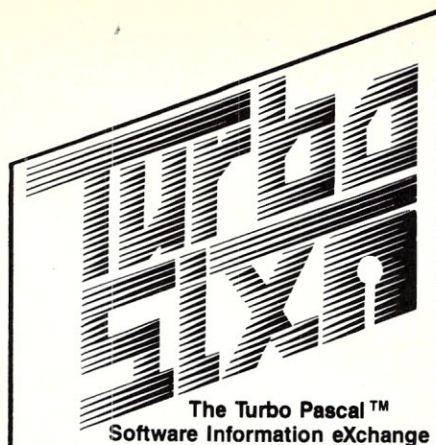
TELEX
643331

For technical assistance or
to request a data sheet, call:

1-203-871-6170



Micromint, Inc.
25 Terrace Drive
Vernon, CT 06066



Megabytes of Turbo Pascal code, including many commercial quality programs, are now available from a single source for the low price of

\$6 per diskette*

*non-member price is \$7.50 per disk

Each diskette is packed with programs and/or routines organized to offer solutions in a selected area of interest. You will save countless hours and learn from the work of your fellow Turbo Pascal programmers.

Some of the selections currently available are:

- 106. DOS Functions & Interrupts
- 504. Telecommunications Programs
- 402. Business Financial Analysis
- 203. Prog. Utilities, Xref, etc.
- 206. Source Code Debugging
- 901. Animation, Sprites w/Graphics
- 603. Turbo Games
- 212. Screen Design Tools
- 201. Windowing & Menu Tools
- 903. Custom Graphics Font Creation

Many more selections available with new titles added regularly. For a complete list send for the TURBO S.I.X. catalog, or better yet join TURBO S.I.X. As a member you will:

- Pay only \$6 per disk. Non-members pay \$7.50.
- Get a MONTHLY newsletter covering what's new in Turbo software, programming tips and other topics of interest to the Turbo Pascal user. Edited by Steve Wood, author of Using Turbo Pascal. (Osborne/McGraw-Hill)
- Get the Turbo S.I.X. catalog and regular updates FREE.

Your one year membership is ONLY \$19.95

The Turbo S.I.X. catalog is available to non-members for \$2.95. Non-Members may purchase diskettes for \$7.50 each.

Overseas orders add \$12.00 for 1 year membership and \$1.50 per disk for Air Mail shipping. Texas residents add 5% sales tax.

(800) 628-2828 Ext. 606

TURBO S.I.X.

P.O. Box 8373, Dept. S1
Waco, Texas 76710

3101 Mitchell, Waco, TX 76708 • (817) 753-2182

```

MOV M,E
INX H
MOV M,D      ;now location 5 contains "jmp xbios"
POP B        ;bc = old bdos entry
POP H        ;hl = start of xbios
INX H
MOV M,C
INX H
MOV M,B      ;now first jmp in xbios is "jmp bdos"
INTRUP
EI
ENDIF
RST 0        ;cp/m warm boot
;Send inline message, terminated by null, to console
;hl must point to the message
MSG EQU $
POP H
MOV A,M      ;a = next char
INX H
PUSH H
ANA A
RZ
MOV E,A
MVI C,CONOUT
CALL BDOS
JMP MSG
MOVEM: MOV A,M      ;move bc bytes from hl to de
STAX D
INX H
INX D
DCX B
MOV A,B
ORA C
JNZ MOVEM
RET
JMPNUM: DS 1      ;will hold # of jmps in bios jmp table
XBPAGE: DS 1      ;will hold page addr of xbios
        DS 64
STACK: DS 2
BUF EQU $        ;spr file goes here first
END

```

LISTING 3 - CXB.C

```

/* cxb.c—checks for xbios modules
Copyright 7/85 N.T.Carnevale
Permission granted for nonprofit use only
*/

#include "printf.h"

#define VERSION "v.1.2 7/16/85\n\n"

#define TRUE 1
#define FALSE 0
#define boolean int

/* handy constants */
#define SIG "xbios installed$"
#define SIGLNTH 16 /* how many characters in signature string */

/* useful structures */
struct xbhdr {
    char jmp, /* replica of xbios module header */
    /* 0xc3 */
    *oldbdos, /* entry to bdos */
    otbloffset, /* how far from ystart to start of local
    copy of old jump table */
    jmpnum, /* how many jumps in table */
    reserved[11], /* unused at present */
    signature[SIGLNTH], /* signature string */
    vstart; /* 1st char of version string */
};

struct cpmhdr {
    /* replica of first few bytes in cpm */
    char jmp1, /* 0xc3 */
    *wboot, /* address of jmp wboot in bios table */
    whatnot, /* user etc */
    drive, /* logged drive */
    jmp2, /* another 0xc3 */
    *bdosloc; /* entry to bdos */
};

char *idstring=SIG;
boolean found=FALSE;

main()
{
    struct xbhdr *xbintro;
    struct cpmhdr *cpmintr;
    char *vers,*descr;
    boolean xbexists;

    printf(VERSION);
    xbexists=TRUE; /* assume there is one */
    /* find start of suspected module */
    cpmintr=0;

```



```

xbintro=(struct xbhdr *)cpmintro->bdosloc;
while (xbexists) {
    /* now see if signature is where it should be */
    if (compare(idstring,xbintro->signature,SIGLENTH)) {
        found=TRUE;
        printf("xbios present at %x\n",(int)xbintro);
        vers = &xbintro->vstart;
        descr = &vers[strlen(vers)+1];
        printf("%s\n%s\n",vers,descr);
        /* try for the next one */
        xbintro=(struct xbhdr *)xbintro->olddbos;
    }
    else xbexists=FALSE;
}
if (!found) printf("no xbios installed\n");
}

boolean compare(s1,s2,num)
/* compare num bytes pointed to by s1 and s2
return TRUE if identical, otherwise return FALSE */
char s1[],s2[];
int num;
{
    int i;
    boolean equal;

    for (i=0,equal=TRUE; (i<num) && equal; i++)
        equal = s1[i]==s2[i];
    return (equal);
}

```

LISTING 4 - RXB.C

```

/* rxb.c—removes most recently loaded xbios module
Copyright 7/85 N.T.Carnevale
Permission granted for nonprofit use only
*/

#include "printf.h"

#define VERSION "v.1.2 7/16/85\n\n"

#define TRUE 1
#define FALSE 0
#define boolean int

/* handy constants */
#define SIG "xbios installed$"
#define SIGLENTH 16 /* how many characters in signature string */

/* useful structures */
struct xbhdr { /* replica of xbios module header */
    char jump, /* 0xc3 */
    *olddbos, /* entry to bdos */
    otboffset, /* how far from vstart to
                start of local copy of
                old jump table */
    jmpnum, /* how many jumps in table */
    reserved[11], /* unused at present */
    signature[SIGLENTH], /* signature string */
    vstart; /* 1st char of version string */
};

struct cpnhdr { /* replica of first few bytes in cpm */
    char jump1, /* 0xc3 */
    *wboot, /* address of jmp wboot in bios table */
    whatnot, /* user etc */
    drive, /* logged drive */
    jump2, /* another 0xc3 */
    *bdosloc; /* entry to bdos */
} *cpmintro=0; /* make it point where it should */

char *idstring=SIG;

main()
{
    struct xbhdr *xbintro;
    char *vers,*descr;
    printf(VERSION);
    /* find start of suspected module */
    cpmintro=0;
    xbintro=(struct xbhdr *)cpmintro->bdosloc;
    /* now see if the signature is where it should be */
    if (compare(idstring,xbintro->signature,SIGLENTH)) {
        printf("removing xbios from %x\n",(int)xbintro);
        vers = &xbintro->vstart;
        descr = &vers[strlen(vers)+1];
        printf("%s\n%s\n",vers,descr);
        interrupt(FALSE); /* disable interrupts */
        remove(xbintro);
        interrupt(TRUE); /* reinstate interrupts */
    }
    else printf("no xbios to remove\n");
}

```

continued on page 85

Does **YOUR** disk Look Like

THIS?



Let

H.D. Tuneup
cleanup your act!

H.D. Tuneup tunes your disk file system, making it better than new. File loads speed along like new. Directory searches are faster than new.

H.D. Tuneup reallocates all of your files to be contiguous. Each file is placed in its proper location, adjacent to its neighbors in the directory.

Disk operations run faster when your files have been Tuned.

H.D. Tuneup relocates your sub-directories into one area of your disk, as close to the root as possible. All deleted entries are actually removed.

Your drive head moves less from directory to directory than when your disk was new. TREE/F will fly when listing all of your files.

H.D. Tuneup requires IBM PC/XT/AT compatibility, 196k, and DOS 2.x/3.x. Fixed disks up to 32mb may be tuned, along with any 5.25" diskette.

NOT COPY PROTECTED!

ONLY \$39.95 plus \$3.00 S/H

SofCap Inc.

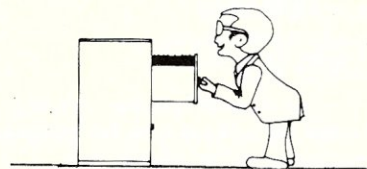
P.O. Box 131

Cedar Knolls, N.J. 07927

Visa (201) 386-5876 M/C

N.J. residents add 6% sales tax.

**For the best performance
from your disk
TUNE IT UP!!**



Visa, Master Card, Checks, Money Orders, or COD only.



dare to compare.

**Don't be fooled.
No hidden cost!
Complete MS-DOS/CPM
Super Turbo**

In keeping with industry trends MSO is bringing our customers high performance P.C. compatibles and accessories. MSO takes the P.C. compatible to maximum performance with its SUPER TURBO. The SUPER TURBO features the V20-8 chip which runs at three times the speed of the IBM-PC XT* and also runs CPM 8080 software.

The SUPER TURBO comes complete with the MS-DOS operating system, Read and Run CPM, full Instructional Documentation, Utility software, **plus for our first 100 customers MSO is offering the Micropro Wordstar Professional Software package. This package includes: Wordstar, Mailmerge, Correctstar, Starindex, Datastar and a G.L. Accounting System.** The SUPER TURBO is a complete turnkey system with everything necessary to plug in and operate.

**USER SUPPORT
HOT LINE**

805/393-2247

All systems carry full 90 day warranty.

CASH PRICE ONLY

Check in advance. Add 3% for VISA/MC. Shipping & handling charges will be added to each order.

For our catalog with complete details and prices, send \$2.00 to:

Micro Supply Organization, Inc.
4909 Stockdale Hwy. #180
Bakersfield, CA 93309

MON. - FRI. 7am - 5pm PST • SAT. 9am - 5pm PST

*IBM is a registered trademark of the IBM Corporation.

15% restocking on returned orders

Super Turbo
Super Price:

\$1299



**The Super Turbo P.C. runs IBM
software and CPM 8080 programs**

- CPU - V20-8 8mHz Super Chip runs *IBM compatible software at 3 times the speed of the IBM-XT and CP/M 8080 software.
- 8087 Math Processor optional
- 256K RAM on mother board expandable to 640K
- ROM 8K Bios
- 6 empty slots for expansion
- 2 serial port one optional with expansion kit
- 1 parallel port
- 1 game port
- Clock calendar with software
- Hi-Res monographics video board
- Floppy controller
- Dual Floppy Drives 360K ea.
- 135 watt XT Power Supply
- 5150 style compatible keyboard
- Hi-Res TTL Green or Amber 12" monitor
- MS-DOS operating system and manual.
- Instructional Documentation and Utility Software
- Assembled and tested in U.S.A.
- **Optional internal 20 meg sub system for Super Turbo add \$499**

**Special printer
pricing with
purchase of
above computer.**

MSO MICRO SUPPLY ORGANIZATION INC.

the alternative.

The SANYO complete word processing system

The alternative for WANG, Xerox or any dedicated word processing system.

At Micro Supply Organization we offer the lowest prices on Sanyo computers and software. With prices like these you can afford the convenience of owning and operating more than one computer. We also offer the User Support Hotline for questions concerning your computer or about software availability. Whether you need one or a dozen computers, Micro Supply Organization is the place to get them!

The MBC 1160 Accounting Software

We carry a complete line of ADS accounting business software. Buy G/L, A/I/R, A/P, Inventory and receive the Payroll module **FREE**. The current retail price is \$398. per module. **Now from MSO you pay only \$99 per module.**

20 meg internal hard drive sub-system for SANYO MBC 1150, 1160 & 1250. **\$649**

USER SUPPORT HOT LINE

805/393-2247

All systems carry full 90 day warranty.

CASH PRICE ONLY

Check in advance. Add 3% for VISA/MC. Shipping & handling charges will be added to each order.

For our catalog with complete details and prices, send \$2.00 to:

Micro Supply Organization, Inc.
4909 Stockdale Hwy. #180
Bakersfield, CA 93309

15% restocking on returned orders



Get more for your money!

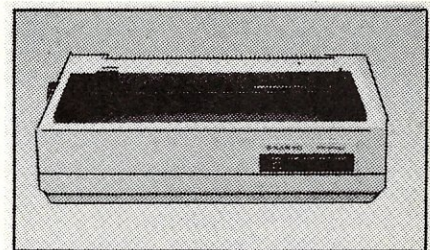
SANYO MBC 1160

Including this **FREE** software:

- Wordstar • Spellstar
- Mailmerge • Calcstar
- Infostar • BASIC

8-Bit Integrated Computer with 640KB Formatted Mini Floppy Disk Capacity

- Z-80A CPU with no-wait mode and large 64KB RAM/4KB ROM memory capacity for fast execution.
- CP/M operating system with editor, assembler and all standard utilities.
- No-glare amber monitor display screen for easy viewing.
- 80-character x 25-line display. 256 characters in 8 x 12-dot matrix cells.
- Two internal double-sided, double-density, double-track 5 1/4" slim-type mini floppy disk drives with 640KB formatted capacity.
- Interfaces for one Centronics printer and one RS-232C port.
- Optional interface for hard disk drive and for external 8" floppy disk drive.



LEGEND 880 OR SANYO PR 5000 Dot Matrix Printer

- 80 char/line or dot-image graphics
- Centronics parallel interface
- Tractors or Friction feed
- Pica, elite, condensed or proportional

Includes printer cable & full warranty

Letter quality Daisy Wheel Printer

- Letter-quality printer with 96-character daisy wheel printing element.
- Accepts paper up to 13 inches in width.
- 14 cps bidirectional printing.
- 10, 12 and 15 cpi printing pitches.
- Centronics parallel interface.
- Optional form tractor.

Includes printer cable & full warranty.

This complete package Computer & Printer lists for \$3194.00

ONLY

\$699

The Scientific Computer User

by A. G. W. Cameron

PMATE

My favorite editor, PMATE, distributed by Phoenix Software Associates (at \$225), has come out in Version 4. I have been using one form or another of PMATE for the last ten years. Every time that I have looked at any other editor written for PCDOS, I have felt crippled because of the lack of power of that editor. To me the test of a good editor is the ability to perform extremely complex manipulations of the text using macros.

An ancestor of PMATE was TECO, a powerful command-oriented editor for use with dumb teletype terminals on minicomputers. Data General had a variation of it for NOVA minicomputers. In the early 1970's, Jonathan Sachs, working for a medical physics project at M. I. T., enhanced this editor, adding considerably to its power and giving it screen display capabilities. He later became the principal author of Lotus 1-2-3.

Sachs' editor migrated to Harvard, where it was further enhanced. Among the enhancers was physicist Paul Horowitz, now well known for having the world's most technically advanced SETI project (search for extraterrestrial intelligence). Paul had a graduate student, Mike Aronson, who took this editor, reorganized and further enhanced it, and rewrote it in assembly language for the 8080 microprocessor. Early versions were named MATE (Mike Aronson's Text Editor), but when Phoenix became the distributor of this CP/M version, they added the "P" in front of the name. Meanwhile, I had become intimately familiar with the Sachs editor, adding about 50 per cent to its length to support the VT100 terminal.

The very first piece of CP/M software that I obtained was PMATE. I disassembled the whole thing, mapped out the command dispatch tables, and made some changes. Meanwhile, the IBM PC entered the scene, and Mike Aronson produced CP/M-86 and MS/PCDOS versions of PMATE. He did it by writing a macro to translate 8080 assembly language into 8086/8088 assembly language. That is an excellent example of the PMATE capabilities. It also meant that when I got an MSDOS machine, I was able to disassemble that version of PMATE and write down the relevant paths through it by inspection, using the CP/M disassembly. In this way I could transport my modifications. I did this with a succession of PMATE versions through version 4.0.

PMATE, Fortran, Plotting, Calculating, T_EX

Enter Murray Sargent III, in the Optical Sciences Department at the University of Arizona, and his small company, Scroll Systems. He devised a text formatter, the Scroller, to format mathematics properly for daisy wheel printers. In the late 1970's he produced small printed circuit boards which inserted into Diablo printers; these intercepted an incoming stream of text and format characters and output the necessary printer control sequences to print the page properly. You could use any text editor, but Murray was a fan of PMATE. Later, for the Victor 9000 and the IBM PC, he did it all in software, merging PMATE with Scroller to form the PS technical word processor. This was the highest quality technical word processor available until T_EX came along. Murray wrote some 9 kilobytes of permanent macros (macro language is extremely compact) to make PMATE a menu-driven editor. The most recent version of PS that I have uses PMATE version 3.37.

PMATE version 4.0 is actually four separate programs, one of which is a limited multitasking environment in which you can run PMATE and another program at the same time. The other three versions of PMATE have different large permanent macro packages, one for word processing, one for Fortran development, and one for C development.

The word processing version is really version 3.37 with most of the PS menus intact (all but the menus for printing) and with some enhancements. The programming development packages take the errors output by the most popular Fortran and C compilers (Microsoft, Ryan-McFarland or IBM Professional Fortran and Lattice C) and place them at the appropriate lines of the source files; they also allow you to move around easily between and within modules of the source file. I do very little C programming have not used that version of PMATE, and since I now do Fortran development with Watfor-77, I have used the Fortran version of PMATE only a little. Many programmers will appreciate these

specialized versions of the editor.

Version 4 of PMATE comes with a configuration file so that you can customize it to your liking, particularly the keyboard locations of the instant commands. There is no configuration file for version 3.37, so it can only be customized with DEBUG, but the appropriate information is not given to allow you to do this. I have done it using my carry-over knowledge of PMATE internals. If you want a copy of my documentation for this customization, send a PCDOS-formatted floppy in a stamped mailer.

For a couple of years Aronson has been rewriting PMATE in C, at the same time modernizing the program so that it will take advantage of all available memory for the text buffer and incorporating multiple display windows. Many of us wait impatiently for this "CMATE". Clearly version 4 is meant only as an interim upgrade.

FORTTRAN

About one and one-half years ago benchmarks by Avram Tetewski at Draper Labs showed IBM Professional Fortran to give faster run-time results than Microsoft Fortran. The IBM package had been developed by Ryan-McFarland, who also marketed the package under their own name. These versions are not identical, since IBM fine-tuned the program to their liking and produced much more extensive documentation. However, the programs give nearly identical benchmarks and can be considered together.

Ryan-McFarland has brought out a major revision in version 2.0. I assume IBM will issue their version of it later. The new version clears up a lot of minor irritants. My own irritant was the lack of a facility for opening a file for appending, but the new version allows this. Such a facility is particularly important when you want to run a program for days on end; if something happens so that the computer hangs or crashes, and you have an open file, then goodbye to all the results. It is a much better technique to open the file for appending when you want to write to it, and then to close it after you have written.

I have made only some very rough measures of performance, but I find that the new Ryan-McFarland version 2.0 runs about 20 to 40 per cent faster than IBM Professional version 1.0. Thus it is now my preferred Fortran for long production runs of a problem. It is typically available for \$399 from discount dealers.

Meanwhile, an important new Fortran product has appeared, Watfor-77 from WATCOM Systems. This is nearly ideal for developing new Fortran code. For many years WATCOM Systems has been in the business of developing language compilers to be used by students learning the language, and they have now brought their expertise with Fortran to PCs.

Watfor does not operate in the usual edit-compile-link-run-edit-etc. cycle. Instead, the program compiles directly into the computer memory and then starts to execute. WATCOM initially supplied me with version 1.2 of the program; later they sent me a beta copy of version 1.4. Version 1.2 required that everything except standard Fortran functions be available as source code, and no EXE file was produced. Version 1.4 is a big improvement in that you can now pull in OBJ files that have been compiled with IBM Professional Fortran as well as routines in libraries intended to work with that compiler, and optionally you can produce EXE files. Watfor also may be run in residence mode; from within the program in interactive mode you can execute any DOS program, including your favorite editor. WATCOM supplies an editor, but it is an old-fashioned one which is line-oriented. Probably the biggest improvement that WATCOM could make at this point would be to develop a good character-oriented editor which is an integral part of the compiler.

On receiving this program I checked it out on some code that I had just spent three weeks developing and for which I was confident that I had found all the bugs. Watfor immediately informed me that there were four variables that had not been initialized. Sure enough, there were four places where I had spelled variables with a 0 (zero) instead of an O—something that I could stare at forever and not notice on the screen. This endeared the program to me right away. There are many checks in the program that are not contained in ordinary Fortran compilers. There are also many extensions beyond standard Fortran, particularly in the realm of controlling the flow of program logic. You should be cautious in using these, however, if you want to transport your source code to other compilers. As a compensation, Watfor gives a warning message when you use these extensions. Watfor has a debugger that lets you set breakpoints at Fortran statements and lets you step through the program by Fortran statements. When the program is halted you can examine the current values of all the variables and change some if you wish.

My rough benchmarks indicate that Watfor compiles into memory in 0.1 to 0.2 of the compile + link time of IBM Professional Fortran. The smaller number applies to cases where the IBM compiler can do a lot of optimization; the larger

number applies where little optimization is possible. This is one of the reasons why I prefer to use Watfor for development. On the other hand, a program run under Watfor will execute a factor of 2 to 4 slower than one compiled with IBM Fortran; the latter number applies when a lot of optimization has been possible. My current advice is to develop under Watfor and do long runs with Ryan-McFarland Fortran; the two programs work together nicely.

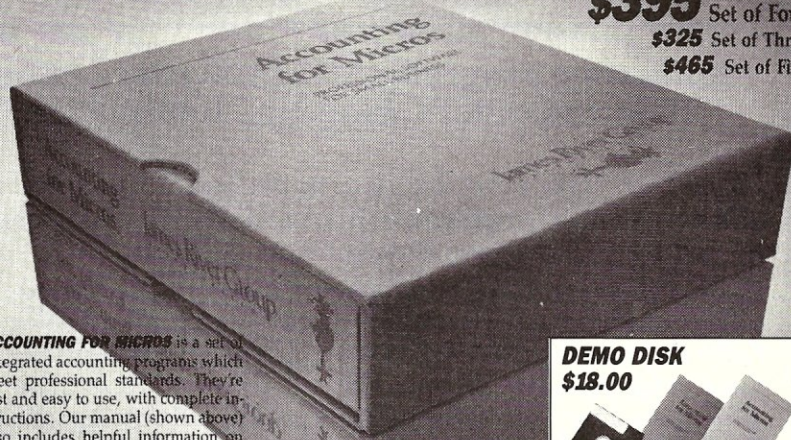
Watfor-77 costs \$295 for an indefinite licence. It is also available on one-year leases with a site license for businesses and educational institutions. The latter version of the code checks that your date is within the allowed range. The section of code that does this is checksummed

and encrypted and the license declares that it is illegal to run with an incorrect date in your machine.

Excellent news for Fortran and Pascal users is the publication of *Numerical Recipes. The Art of Scientific Computing*, by W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling (Cambridge University Press, xx + 818 pp., \$39.50, produced using T_EX). The authors state, in their introduction, "this book is unique, we think, in offering, for each topic considered, a certain amount of general discussion, a certain amount of analytical mathematics, a certain amount of discussion of algorithmics, and (most important) actual implementations of these ideas in the forms of working computer rou-

ACCOUNTING FOR MICROS

\$395 Set of Four
\$325 Set of Three
\$465 Set of Five



ACCOUNTING FOR MICROS is a set of integrated accounting programs which meet professional standards. They're fast and easy to use, with complete instructions. Our manual (shown above) also includes helpful information on bookkeeping and computers.

GENERAL LEDGER \$125
Allows up to 1,000 accounts & 1,000 transactions/month. Retains month-end balances for Last year, This Year and Forecast. Includes Cash Disbursements, Cash Receipts and General Journals. Reports include Balance Sheet, Income Statement, Annual Summaries and Journal Reports.

ACCOUNTS RECEIVABLE \$125
Allows up to 2,500 customers and 1,000 invoices per month. Invoicing can access Inventory Module. Keeps customer names and addresses. Invoice prints on plain paper or any pre-printed form. Statements can be printed at any time.

INVENTORY \$125
Allows up to 4,000 parts. Keeps 3 month history of unit sales as well as year to date. With AR, can be used as point of sale system (prints invoices, handles cash). Reports include Inventory Value and Stock Report, Internal and Customer Price List.

ACCOUNTS PAYABLE \$125
Allows up to 500 vendors and 600 invoices/mo. Records invoices and handwritten checks. Prints computer checks on any pre-printed form. Keeps vendor names and addresses.

PAYROLL \$125
Will handle up to 100 employees with eight deductions per employee. Deductions may be determined as fixed dollar amounts or percentages, or referred to a table for automatic look-up. Tax tables are easily entered, or purchased separately. Prints checks and W2's.

SET OF FIVE \$465
SET OF FOUR \$395
SET OF THREE \$325

RUN ON MOST CPM AND MSDOS

Apple CPM	IBM PC, XT, AT	Sanyo (all)
Columbia	Kaypro (all)	Tandy (all)
Compaq	Morrow (all)	TeleVideo
Corona	Oxborne (all)	Zenith 100 & 150
Eagle (all)	Panasonic	8" CPM
Epson QX-10	Radio Shack CPM	Other compatibles

DEMO DISK \$18.00

Try all 5 programs above (GL, AR, AP, IN, PR). Order our DEMO DISK for \$18.00 (includes shipping). Condensed versions of the programs give you the "feel" of data entry and access. Includes sample reports and instructions. Specify machine.

TMAN \$125
The "Catch-All" program. Files any type of information for quick access. Name or subject oriented with 15 lines of notes per name. Use TMAN as a mailing list, filing system, notebook, etc. Can be used alone or with data from our other programs.

Try **TMAN DEMO \$16**

HOW TO ORDER: Please specify machine and disk format. You can pay by check, by VISA or MasterCard (we need your expiration date and card number), or by UPS COD (add \$2.50 COD charge). Our price includes shipping. Minnesota residents, add 6% sales tax. We ship most orders the same day.
or ORDER BY PHONE: 612-339-2521

James River Group
INCORPORATED

(612)339-2521



125 North First Street
Minneapolis, MN 55401

Instant-C™: **The Fastest Interpreter for C**

**Runs your programs 50 to 500 times faster
than any other C language interpreter.**

Any C interpreter can save you compile and link time when developing your programs. But only **Instant-C** saves your time by running your program at compiled-code speed.

Fastest Development. A program that runs in one second when compiled with an optimizing compiler runs in two or three seconds with **Instant-C**. Other interpreters will run the same program in two minutes. Or even ten minutes. Don't trade slow compiling and linking for slow testing and debugging. *Only Instant-C will let you edit, test, and debug at the fastest possible speeds.*

Fastest Testing. **Instant-C** immediately executes any C expression, statement, or function call, and display the results. Learn C, or test your programs faster than ever before.

Fastest Debugging. **Instant-C** gives you the best source-level debugger for C. Single-step by source statement, or set any number of conditional breakpoints throughout your program. Errors always show the source statements involved. Errors always show the source statements involved. Once you find the problem, test the correction in seconds.

Fastest Programming. **Instant-C** can directly generate executable files, supports full K & R standard C, comes with complete library source, and works under PC-DOS, MS-DOS, or CP/M-86. *Instant-C gives you working, well-tested programs faster than any other programming tool.* Satisfaction guaranteed, or your money back in first 31 days. **Instant-C** is \$495.

Rational
Systems, Inc.

P.O. Box 480
Natick, MA 01760
(617) 653-6194

times." The goal of the book is to include everything up to, and even a bit about, partial differential equations. Surprising as it may seem, the discussion includes a touch of wit now and then. This approach is far superior to the standard collections of scientific subroutines, of which several are now available for PCs, since it provides a good guide to when you should use one technique and when another.

Commented Fortran source code is provided throughout. The Pascal equivalents are collected toward the end of the book. The Fortran and Pascal source code, without comments, is available on separate diskettes for PCs (\$19.95 each). There are also example diskettes (\$19.95), and somewhat overpriced *Example Books*

(\$18.95) in softcover for each language.

PLOTTING

Fortran libraries of plotting subroutines have for the most part concentrated on screen displays and output to pen plotters. I have found this frustrating because dot matrix printers such as the FX-80 and its compatibles have the sufficient resolution to produce publishable graphs. But until recently none of the software products have taken advantage of this resolution. Now such a product has appeared; it is SciPlot, from MicroGlyph Systems (\$59.95). I have been very interested in following the progress of this product over the last few months as it has been contin-

ually enhanced by its author, C. L. Vote, in part due to my prodding. It started out as a CP/M product.

SciPlot has the usual commands that one finds in scientific plotting packages, although they are not quite compatible with CalComp routines used on mainframes, so that if you want to transport a program a little rewriting will be necessary. Like the better plotting packages, it makes use of subsets of the Hershey vector character set, and you can use proportional spacing. Mr. Vote has also included several interpolation routines for manipulating data; if these are not good enough for your application you could use others in *Numerical Recipes*. A supplemental package gives source code for the subsets of Hershey characters and Fortran tools to manipulate them. This allows you to change characters or design new ones. Since the Hershey character set is in the public domain, you could also make up sets of missing Hershey characters, although you would have to get the vector listings separately.

SciPlot plots on CRT displays and on a few hard copy devices. Initially a color display was required, but the Hercules monochrome graphics card is now supported. Dot matrix printers are the IBM Graphics/Proprinter and the Epson FX185/FX-80/MX-80/etc. In addition, the Apple LaserWriter can be used as a high-resolution graphics device; Mr. Vote has just begun to probe the potentialities of this device, since he just uses the vector plotting capabilities and does not use the internal character sets (however, this is one way to use the Hershey characters with this laser printer). The LaserWriter output looks very nice; I would recommend that line widths be scaled to character sizes when Mr. Vote makes his next revision. The dot matrix plots are adequate for publication in most professional journals.

MATRIX CALCULATOR

Matrix Calculator from SoftTech Inc. (\$59.95) is a general purpose interactive math package. It is available in a plain stand-alone version, an 8087 version, and a memory-resident version called RESIDENT. Most of the operations are intended to be done on matrices, which can be quite large in size. However, the simplest matrix is a single number, and I found that doing ordinary arithmetic and finding standard functions of numbers with this program was intuitive.

In addition to the matrix operations, which include solutions to sets of linear equations, a variety of numerical integration, statistical, and other functions such as gamma functions are available. The package is very powerful. The input and output can be stored as disk files, so there will be many cases in which people will prefer to work with data interactively rather than

to design Fortran programs to do so. The manual has a very large number of spelling errors, but I find that these detract rather little from the comprehension of the procedures. Very definitely worth while.

TEX NEWS

See my previous columns for extensive discussions of \TeX . Here is an update. Both \MicroTeX (Addison-Wesley, now \$295 without a printer driver) and \PCTeX (Personal \TeX , now \$249 without a printer driver) have released version 1.5 of \TeX . The main advantage of this for microcomputers is that the memory allocation procedures are revised to be more efficient, so that some unusual conditions which might have overloaded memory in previous versions of \TeX will now be handled properly. Personal \TeX is also distributing a program to convert the dot matrix pixel font files to a more compact form, which takes up only about 35 per cent of the space. This will be a big boon to hard disk users.

Version 2.0 of \TeX will differ from 1.5 only in the sign-on message, but with version 2.0 will come the "final" set of \TeX fonts, which are called "Computer Modern", in place of the existing distributions of "Almost Modern" fonts. Thus I suspect that there will be a delay of many months before \MicroTeX and \PCTeX are issued in version 2.0 with all these new fonts. This change

Addresses

Addison-Wesley Publishing Company, Reading MA 01867. (617) 944-3700.

American Mathematical Society, P. O. Box 1571, Annex Station, Providence, RI 02901-1571. (800) 556-7774.

Cambridge University Press, 510 North Ave., New Rochelle, NY 10801. (914) 235-0300 in NY and Canada, (800) 431-1580 elsewhere.

MicroGlyph Systems, P.O. Box 1066, East Arlington MA 02174.

Personal \TeX , Inc., 20 Sunnyside, Suite H, Mill Valley CA 94941. (415) 388-8853.

Phoenix Software Associates, 1416 Providence Highway, Suite 220, Norwood MA 02062. (800) 344-7200.

Ryan-McFarland Corporation, 609 Deep Valley Drive, Rolling Hills Estates, CA 90274. (213) 541-4828.

Scroll Systems, Inc., 6930 East Acoma Place, Tucson, AZ 85715. (602) 885-1633.

SoftTech, Inc., 18505 W. 8 Mile Rd., Detroit, MI 48219. (313) 544-8544

WATCOM Systems Inc., 415 Philip Street, Waterloo, Ontario, Canada N2L 3X2. (519) 886-3700.

marks the coming to maturity of Donald Knuth's **METAFONT** program, which is used to design characters in new fonts. I

look forward to the time when this program will be available on PCs; three of his related books describing it are to be published by Addison-Wesley this summer. These are *The METAFONTbook*, **METAFONT: The Program**, and *Computer Modern Typefaces*.

Large \TeX macro packages have now become available. These include the final version of \LaTeX , a major package which relieves users of the necessity of giving a lot of "set-up" commands to \TeX ; the manual, *LaTeX*, by Leslie Lamport, is available from Addison Wesley (\$18.95). Also now available is \AMS-TeX , which incorporates styles appropriate to preparing articles for publication in journals of the American Mathematical Society. The manual, *The Joy of TeX*, by M. D. Spivak, is available from the Society (\$34).

\TeX is now available for the Macintosh. \MacTeX is actually version 2.0 of \TeX , and so it will have the Computer Modern fonts distributed with it. It was developed by David Kellerman and Barry Smith in Portland, Oregon, and will be marketed by Addison-Wesley (\$495). This program will have the capability of including MacPaint and MacDraw pictures in \TeX documents. μ

A. G. W. Cameron is Professor of Astronomy at the Harvard-Smithsonian Center for Astrophysics, 60 Garden Street, Cambridge, MA 02138.

MAIL TO:

P.D.S.C. Co. (est. 1983) Suite L3, 33 Gold St. NYC 10038 212 732-2565
West Coast Orders May Use Our Napa Phone 707 224-8931 4-10 p.m.

CP/M SOFTWARE Price List/Order Form

We recently acquired 80% of Osborne Computer Corp.'s Software stock, in a liquidation auction and can offer regular commercial grade software at prices far under usual wholesale. Most programs are copyable, and are in Osborne SD format, which we understand is readable by Commodore 128, Kaypro, Zorba and many others. Some of this software may be licensed for use on the Osborne Computer; you may want to obtain a second machine license or update from the publisher directly. Please ask about quantity/dealer pricing. We also have Osb manuals/parts. Circle items ordered.

- | | |
|---|--|
| Class A
\$17.00
each | Documate plus—indexes and builds table of contents. Deadline—solve murder game. ZORK-1—Adventure game. Enumerator—adds line numbering to text files/Wordstar. Starcross—Galaxy game. Grammatick—checks grammar/syntax. Footnote—for text files/books. Math—adds calculator to W.S. |
| Class B
\$28.00
each | StartPac—tape tutorial intro to CP/M WS, Scalc, Basic. Forth v2.0—complete language for programming. MicroLink. DBase 11 tutorial set. Disk Doctor—repair kit for crashed files. Milestone—project mgr.. Mailman—for lists. Spellguard—corrects. Osborne-1 Diagnostic testor. Fortran. DD CP/M Utilities upgrade. BASCOM (No Manual). |
| Class C
\$39.00
each | BSTAM—super modern, great for multi-file transfers. SuperSort—easy data manager, very capable. Personal Datebook. Wordpac: writer's packages for use with Wordstar, Orig. \$295.00 including Grammatick Spellguard, Math, Documate plus, Footnote. DataStar. Real Estate Mgr. SuperCalc. MBASIC. |
| Class D
\$95.00 ea. | Osborne 1 5 Pac including Microsoft BASIC, SuperCalc, and Wordstar v2.24 with Manual—Complete text editor. Personal Pearl—menu driven database manager, already set up. Any 3 of class C, or 4 of class B, or 8 of Class A. |
| Class E
\$175.00 ea. | DBase 11 w/Zip—the original relational database manager. Executive set: Wordstar, Supercalc, C/MBASIC, CP/M 3.0, Personal Pearl. All titles are copyrighted by their respective publishers.
OSB: Field Service Guide \$75.00. Exec. tech parts A&B \$75.00 (Excl. Bios)
OSB: Gray Owner's Guide, Wordstar, Scalc, BASIC \$20.00. Exec. 4PAC Books \$50.00. |

We also maintain a complete library of CP/Mug and SIG/M Public Domain CP/M Software. Presently about 90 meg/20,000 programs. Catalog \$4.00. 1 volume on 2 DD SS 0-1 Disks \$10.00. Post/Handling \$3.00 order.

We have Public Domain Libraries for TRS-2/CoCo, Commodore 64/Vic 20, Apple II e.c. plus, Macintosh, and I.B.M.-PC/Compatibles. Ask for Catalog. Please advise us of your needs. We will not be undersold. Terms: All sales are final due to closeouts.

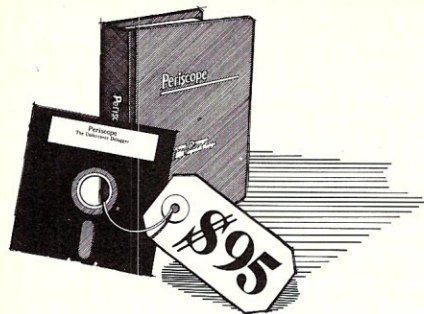
Please Advise (Card #) _____ Amt. Enc. _____

Regards
P.D.S.C.

Name/Phone/Computer _____

Address/Zip _____

Periscope Is On Sale!



"A marvelous debugging tool..." says Peter Norton.

SUMMER SALE. Through August 31, the new software-only Periscope II-X is yours for just \$95! Periscope II, normally \$145, is only \$125!

NEW MODEL. Periscope I includes both a break-out switch for "spontaneous" debugging and a RAM board to protect the debugger from programs that overwrite memory. Periscope II includes a break-out switch that doesn't require an extra slot to install. And the new Periscope II-X is Periscope II without the break-out switch. So you can now get Periscope with no hardware at all!

NEW RELEASE. The new version 2.1 enhances all models with many of the features users have requested. Call for details.

CHECK OUT THE REVIEWS.

"Periscope strikes a balance of power and features that will please serious PC programmers."

—Jeff Duntemann, PRODUCT OF THE MONTH, PC Tech Journal, 1/86

"Periscope's diverse features, affordable price, and portability place it in a class by itself." —Ward Christensen, "Breaking out with Periscope," PC Tech Journal, 3/86

"Periscope represents the finest software debugger available in its class." —Andrew Fried, Computer Shopper, 4/86

REQUIREMENTS. An IBM PC, XT, AT or close compatible; 128K RAM; DOS 2.0 or later; one disk drive; and an 80-column monitor.

Periscope II-X, Software Only \$ 95*

Periscope II, with Switch \$125*

Periscope I, with Board & Switch . . \$295

*Prices good through August 31, 1986

Take advantage of the low sale prices while they last . . . Order your Periscope today! Call toll-free:



800/722-7006



30-Day Money-Back Guarantee

Shipping—\$3 UPS ground; \$5 COD; \$8 Canada; \$24 Foreign. We accept qualified PO's in the U.S.

Get Your Programs Up and Running;

UP PERISCOPE!

Data Base Decisions • (404) 256-3860
14 Bonnie Lane • Atlanta, GA 30328

XPIP

by Edward J. Joyce

Extended Utilities For MS-DOS & CP/M

MS-DOS and CP/M resemble houses without furniture. They satisfy a basic need, like empty houses provide shelter, with few concessions for comfort and convenience. Consequently, most computing users invest in accessory software designed to improve one's lot while living within the domains of these operating systems.

One of the best bargains in accessory software for the MS-DOS and CP/M environment is an extended utility called XPIP. This \$29.95 program fills many of the gaps found in the bare bones operating systems. The developers of XPIP call it an integrated system utility. Functionally, it serves as a super file handler that performs basic file maintenance plus dozens of additional tasks.

XPIP operates in two modes: as a one-line command with options or from a menu of nearly twenty-five selections. If a question arises regarding a selection in the menu mode, a single keystroke produces help information. The help file spans 50K on disk, which gives a clue as to the thoroughness of the context-specific information.

The command selection covers standard operations such as file renaming, copying and deleting. The advantage of using XPIP for file deleting instead of MS-DOS's ERASE or CP/M's ERA command lies with XPIP's options. Each command selection is coupled with about ten options depending on the nature of the command. For example in deleting, you may first sort the file directory by name, type (or extension) or size, mark the files to be deleted individually, or specify

groups of files to be deleted through wildcard selectors in the file specifications.

The MS-DOS version supports additional options for file selection criteria. You can select files according to their creation times such as before or after a given date.

Other options pertain to output. The displayed output may be printed with a selectable number of lines per page. Similar control exists over the screen. An initial screen definition step configures the software for any terminal. During execution of XPIP, the speed of scrolled displays may be increased or decreased by pressing designated keys.

XPIP shows its strength as a super tool in the areas of searching, editing and maintaining directories. String searches may be conducted within a file or across an entire disk. XPIP displays each occurrence of the requested text along with the file name and line number within the file.

The editor alone is worth the price of the entire package. It operates on 128-byte sectors within a file. Any sector may be accessed by sequential scanning or by entering the record number. The sector contents are displayed in both ASCII and hexadecimal form. Characters can then be modified and the sector subsequently updated on disk.

Maintaining MS-DOS directories becomes a snap under XPIP. The program makes and removes directories and shows directory tree structures. A shortcut for moving files between directories may rank as the program's greatest timesaver. Instead of physically copying a file from one directory to the other and erasing the original, XPIP simply adjusts the directory indexes. The adjustment entails a fraction of the time required to make a physical copy and the user needn't worry about allowing for sufficient space to temporarily accommodate the copy.

XPIP embodies numerous other commands including file comparison and math calculations (in binary, octal or hexadecimal). If you had the time, you could probably piece together some of its functionality from public domain software. But you would be hard pressed to find all of its commands within one package accompanied by a sixty-page user's manual. At \$29.95, this is one software accessory that can't be beat. (µ)

Ed Joyce is author of Modula-2: A Seafarer's Manual and Shipyard Guide, Addison-Wesley, Reading MA 1985.

XPIP PRODUCT INFORMATION

System Facilities, Inc.

Box 7079

Charlottesville, VA 22906

(804) 977-5245

Hardware Requirements



128K RAM - MS-DOS

44K TPA - CP/M

Price \$29.95 + \$3 shipping & handling

RP/M2™ creates CP/M®2.2 compatible IBM PC

1. Remove the 8088
2. Install the NEC μ PD70108 (V20)
3. Boot PC RP/M2

The μ PD70108 chip is a fast 8088 that also executes 8080 machine code. PC RP/M2 is an **operating system**. Either standalone or with MSDOS present, PC RP/M2 provides the solid base of a genuine operating system reliably distinct from the facade created by an MSDOS interface. All 2.2 features are supported with 56.5k TPA, plus file date and time stamping, fast virtual disk, iobyte redirection, terminal cursor positioning, color console display. DOSDISK accesses MSDOS drives; MODEMPC transfers files via serial port; SETDISK redefines a drive to any of over 80 formats, allowing PC RP/M2 to directly process existing CP/M files in their native form. Two system disks plus 105 page user and technical manual \$129. Shipping \$5 (\$10 nonUS)  



**Micro
Methods, Inc.**

118 SW First St. - Box G
Warrenton, OR 97146
(503)861-1765

continued from page 77

```
boolean compare(s1,s2,num)
/* compare num bytes pointed to by s1 and s2
   return TRUE if identical, otherwise return FALSE */
char s1[],s2[];
int num;
{
    int i;
    boolean equal;

    for (i=0,equal=TRUE; (i<num) && equal; i++)
        equal = s1[i]==s2[i];
    return (equal);
}

interrupt(choice)
/* if choice TRUE, enables interrupts
   else disables them */
boolean choice;
{
    /* not implemented in this version,
       since my bios doesn't use interrupts.
       --use inline code if needed */
}

remove(xbmod)
struct xbhdr *xbmod;
{
    int i;
    struct jumpintr {
        char jump; /* the 0xc3 itself */
        *dest; /* the target address */
    } *xbjmptbl,*biosjmptbl;

    /* pick up start of next xbios module
       from one about to be removed */
    printf("new bdos entry will be %x\n", (int)xbmod->oldbdos);
    cpmintro->bdosloc=(char *)xbmod->oldbdos; /* fix jmp
       bdos at 5 */
    /* next copy "old jump table" from module into bios */
    xbjmptbl= (struct jumpintr *)(&xbmod->vstart +
        (0x00FF & xbmod->otbloffset)); /* mask out hi
       bits of offset */
    biosjmptbl=(struct jumpintr *) (cpmintro->wboot);
    for (i=0; i<xbmod->jmpnum; i++)
        (biosjmptbl++)->dest=(xbjmptbl++)->dest;
}
```

AT LAST: Professional Typesetting Capability For PC Users

With **PCTEX™** — the best-selling full implementation of Professor Don Knuth's revolutionary typesetting program **TEX**.

FINEST Typeset Quality Printing From:

dot matrix laser phototypesetter

$$\sum_{i=1}^{\infty} \frac{1}{i} \begin{pmatrix} a_{11} & \dots & a_{1n} \\ a_{21} & \dots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \int_{-\infty}^{\infty} e^{-x^2} dx$$

WIDEST Range Of Output Device Drivers:

- Epson FX, LQ
- HP LaserJet*
- Toshiba
- Apple LaserWriter
- Corona LP-300*
- APS-5 phototypesetter
- Screen preview, with EGA or Hercules card

MOST COMPLETE Product Offering:

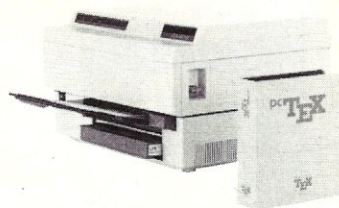
PC_{TEX} (not copy protected) includes the following:

- Our specially written *PC_{TEX} Manual*, which enables you to start using **TEX** right away.
- Custom "macro packages" that provide formats for letters, manuals, technical documents, etc.
- The **L_ATEX** document preparation system, a full-featured macro package for preparing articles, books, reports, etc., and **L_ATEX User's Manual**.
- **A_MS-TEX**, developed by the *Amer. Math. Society* for professional mathematical typesetting.

Site licenses, volume discounts, and interfaces to PC Paintbrush, PC Palette, FancyFont and Fontrix are also available.

PRICED FROM ONLY \$249.00!

(Printer drivers and interfaces additional.)



**Laser printer,
fonts & software
from \$2995.00**

For IBM PC/XT, AT or compatible, DOS 2.0 or higher, and 512K RAM. Hard disk required for printer drivers and fonts.
*HP LaserJet and Corona require additional interface boards.


For more information call or write:

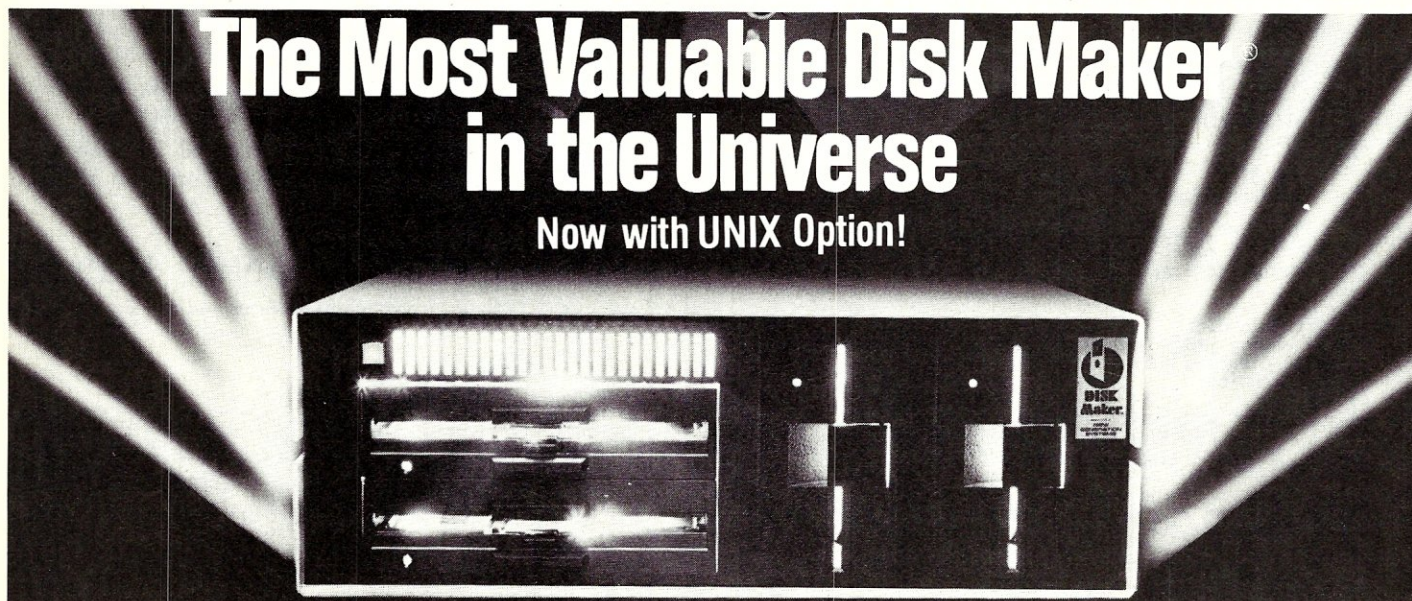
Personal TEX, Inc.

20 Sunnyside, Suite H, Mill Valley, CA 94941 (415) 388-8853

This ad, with space for the photograph, produced by PC_{TEX}. Typeset on the Epson FX80, the Corona LP-300 laser printer, and the Autologic APS-5 phototypesetter.

TEX is a trademark of the American Mathematical Society. Manufacturers' product names are trademarks of individual manufacturers.

object \$49	BASIC WINDOWS	asm source \$149 *
Create "instant" pull down menu's or pop-up help windows using any color combo's desired. Basic Windows uses a smart window buffer manager which does not use available BASIC memory! Also allows async operations for a dozen simultaneous windows. Screen save/restore, window create/restore/toss, window print using word wrap & other controls, Input string\$ from window, clear/scrollup/ scrollldn/clearEOL&EOS. Includes 2.5k code, doc, test program. For IBM & Compaq BASICA or IBM & MSQUICK compilers on Dos 2 or 3.		
object \$59	C DATA ENCRYPTION	c source \$249 *
A complete security system using DES (Data Encryption Standard) per U.S. government standard FIPS PUB46 written in Microsoft C. The 3 modes are supported: Electronic code book, cipher feedback, and cipher block chain. Also included is data compression and telecommunications formatting for transmission and storage via any network medium. Send your electronic mail encrypted via Telemail, MCI mail, Easylink Telex, or BBS ?!!		
object \$99	COMMx	consolx \$69 option
COMMUNICATIONS eXchange provides multi-terminal emulation, multi-protocol file transfers, unattended control macros and multi-site dial & parameter directory. Emulates VT100/102/52, IBM3101-1x, WYSE50, TV950, HP2624/45, ADM3/11, ADDS R60 (all character modes). Terminal modes support full color controls with optional 25th status line showing time and caps/num lock status. File XFER for ASCII, KERMIT, XMODEM, and COMMx (faster than KERMIT using COMMx/Mainframe). CONSOLX option provides remote PC operation.		
 HAWKEYE GRAFIX Inc	3415 Hyde Park Drive Clearwater, FL 33519 call => 813-786-8161	* discounted thru Sept 1986



Download fast, read over 200 formats easily, reformat rapidly

The more disk formats you work with, the more our Disk Maker® system saves time and money by reading and/or writing disks in any of over 200 formats. No modems, no patches, no other special software necessary.

Disk Maker II Plus is a complete, stand-alone system featuring both 8" and 5 1/4" disk drives with options for 3 1/2" and IBM PCAT drives—up to 12 total drives. With its 6 MHz processor and RAM memory drive, Disk Maker II Plus is fast! Hardware options include a 15 or 20 Mb hard disk, an 8086 MSDOS coprocessor and memory expansion.

Optional software includes word processor and Compu-graphic typesetter formats and the MicroDrive Tester for drive alignment testing.

Supported with comprehensive manual, 6 month warranty and continuing software update program.

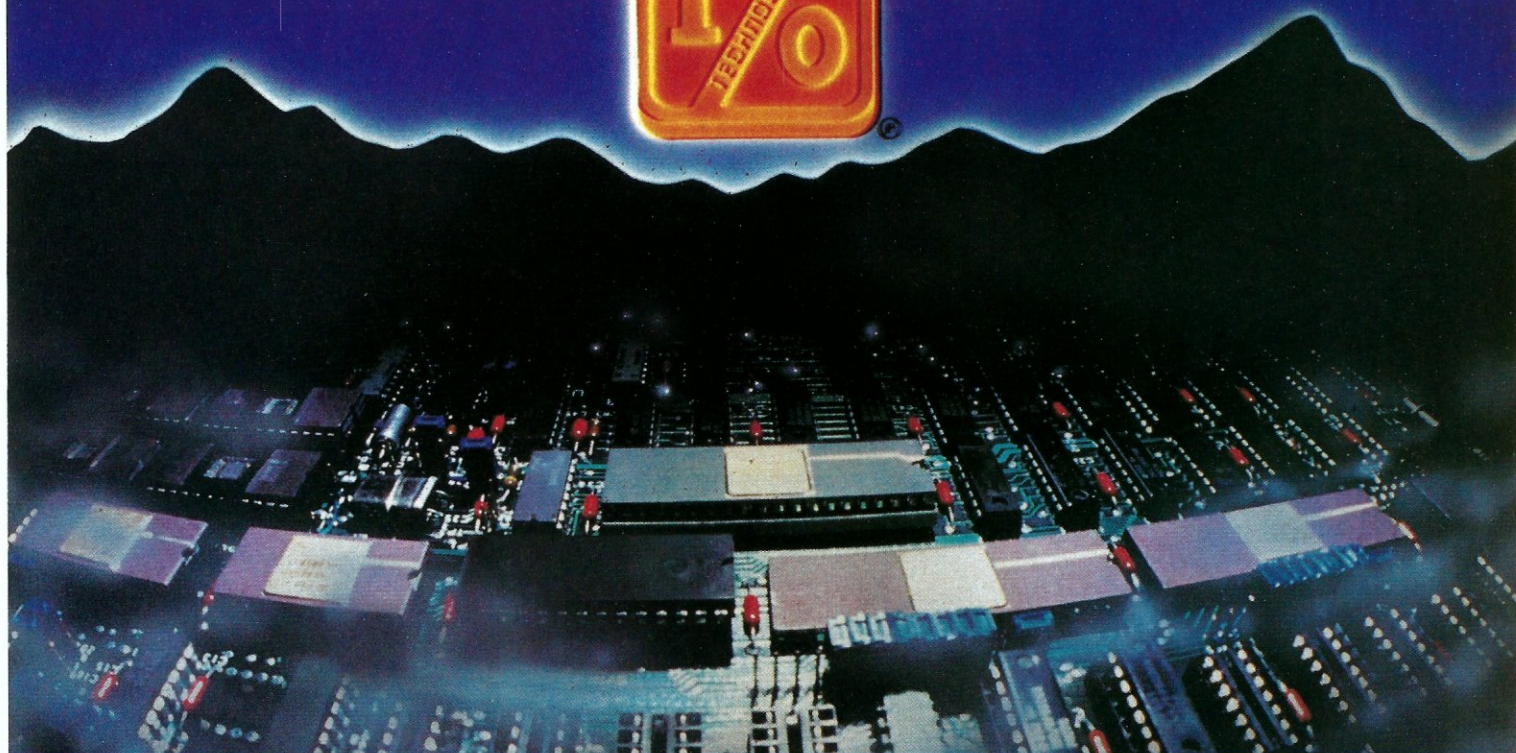
Disk Maker I is an alternative for existing S-100 systems (CompuPro, Zenith Z-100, North Star, etc.) to provide the same functions as Disk Maker II Plus.

Write or call today for information on how Disk Maker® can save you time and money.

Jerry Pournelle wrote about Disk Maker®: "It's easy to install, easy to use, and darned near unique. If you need one, you need it bad." (BYTE, July 1984)

**NEW
GENERATION
SYSTEMS**

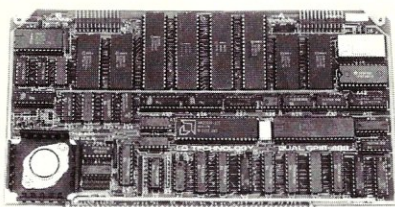
1800 Michael Faraday Drive, Suite 206, Reston, VA 22090
(703) 471-5598 Order Line: (800) 368-3359
Dealer inquiries welcomed.



INPUT/OUTPUT TECHNOLOGY, INC.

25327 Avenue Stanford, Unit 113, Valencia, CA 91355 • [805] 257-1000

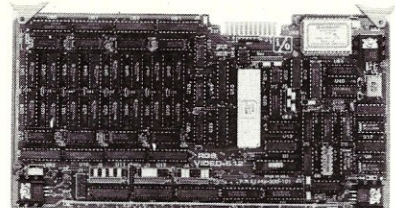
Uncompromising Additions to your S-100/IEEE-696 BUS



DUAL GPIB-488 INTERFACE BOARD

A Stand-Alone, Independently Controlled Dual Channel IEEE-488 I/O Processor. Interface Activity Modes for Controller-in-Charge, Controller Assigned or Terminal Bus Slave, and all Interface Functions are handled transparent to Host System CPU through an on-board CPU and DMA controller. User Friendly operation.

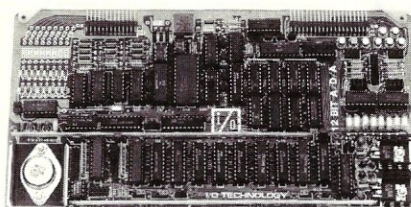
A&T, P/N 52748-800-102



RGB COLOR GRAPHICS BOARD

Programmable resolution up to 512 x 512 pixels with 4 local video planes and on-board graphics processor. Color mapper allows 16 colors from a palette of 4096. Light pen input. Plus more ...

A&T, P/N 52748-300-101

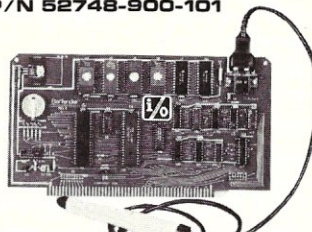


12-BIT A-D CONVERTER BOARD

8 Channel A-D: 12 microsec. Conversion, 50KHz Sample Rate, Programmable Gains, Offset and Diff./Single Modes.

8 Channel D-A: 2 microsec. Settling, Bipolar V or Unipolar I Output. Programmable Reference levels, Dual-Ported Channel Refresh RAM. **16/8-Bit Data Transfers** via I/O or Memory Mapped

A&T, P/N 52748-900-101

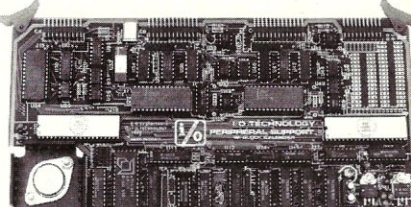


BAR CODE PROCESSOR BOARD

The BarTender is a stand-alone I/O Processor that reads and prints most common Bar Codes. Includes bi-directional reading, wand interface, clock/calendar with battery. Extensive documentation and software.

A&T, 52748-500-101 Without Wand

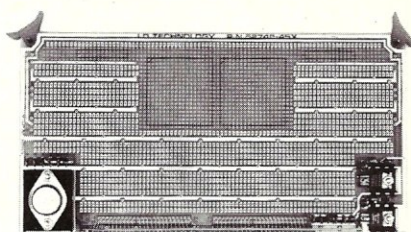
A&T, 52748-500-201 With Wand



PERIPHERAL SUPPORT BOARD

Two Serial SYNC/ASYN Port with RS-232, TTL or Current Loop Outputs, three 8-Bit Parallel Ports, three Timers, Real Time Clock/Calendar and Response Programmable Interrupt Controller. Small Proto Area with +5 and $\pm 12v$.

A&T, P/N 52748-150-101



MULTI-PURPOSE PROTOTYPING KIT

Industrial Quality with Plated-Thru holes for Wire-Wrap or Solder projects. Complete with +5, $\pm 12v$ Regulators, Bus Bar, Filter Capacitors, and Manual.

P/N 52748-450

ALSO AVAILABLE: MULTI-FUNCTION I/O BOARD, SMART PROTOTYPING KIT, 128Kx8/64Kx16 STATIC RAM MODULE

SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE.

The Data Base Forum

by Nelson T. Dinerstein

Reflex was developed by Analytica who failed at marketing the product and sold it to Borland. Borland cut the price to a fraction of the original and in the first two weeks over 15,000 copies were sold.

Reflex provides many of the features that you would expect to find in a simple database management system. It also provides some rather nice data analysis tools. Note that these tools are not the standard statistical tools, although some of the basic statistical functions are provided.

You can, as expected, define a database file structure, enter data, modify, delete data, and produce reports. Each user application is limited to one database file; it is not possible to link files together. Since you are limited to one file, the best you can do (in most cases) is to design a file that is in the first normal form. In particular, the typical file will contain information about two or more real world objects. For example, consider a file that contains sales information. To store a complete record, the file needs to contain information about each of the following: the salesperson, the item sold, the purchaser, and the specific sale information, i.e., the date of the sale, the quantity purchased, etc. This can cause two problems: large amounts of data are duplicated (potentially using too much space), and the data can be difficult to maintain. Reflex allows you to designate a field as REPEATING TEXT, if you find that there is a considerable amount of duplication. It appears that REPEATING TEXT reduces the amount of disk space required to store such a file, but true factoring has not been implemented. As a result, if you wish to change a value in a Reflex database file, you must manually change it in all places where it occurs. Since Reflex does not contain a programming language, the user must enter all the keystrokes necessary to perform the desired operations. Because of the problems with the maintenance of the data and because of the lack of a programming language, Reflex is probably not suitable for the development of professional quality information systems. Nevertheless, I believe Reflex has its place in the scheme of things.

Some of the features I like about Reflex are:

1. Reflex is relatively easy to use, since all of the functions are invoked through the use of pop-up menus. Even if you don't know how to use Reflex, you can work your way through a typical application by just examining the menus

A System Builder's Review Of Reflex

until you find the function that you want.

2. The functions you need to enter, store, manipulate, and report data have been included. In addition, a powerful yet easy to use tool to select records has been integrated with a number of functions. You can specify the selection criteria and then use it either to visit the individual records one at a time or you can use it to filter out records that either do or do not satisfy the selection criteria. After the records have been filtered out, they are easily restored with a single keystroke.
3. The report writer is unusually easy to use, powerful and flexible.
4. Working data can be displayed on the screen simultaneously in up to three different formats or views: as a list, a form, or a graph. The list view presents the data in tabular format, the form view presents the data one record at a time as a document, and the graph view presents the data in graphical form. You can move from one window (view) to another, add new windows (up to three), delete windows, and change the size of individual windows. You can even fill the screen with one of the windows and then shrink it back down again when you are through with it.
5. A reasonably powerful set of functions is included. You can mix functions, names of fields, constants, and arithmetic operators to create powerful formulas that can be used in a variety of places. The formulas can be used to create computed fields in the data base or a report, help you to select records from the file for observation, help you to select records from the file for observation and analysis, and perform, summary computations in a report.

6. The analytical features of Reflex are provided through the use of selection formulas, graphics and Crosstab. Crosstab is an amazingly powerful tool for the analysis of your data, allowing you to perform each of the following operations, singly or in combination:

- A. Select the data (records) to be examined.
- B. Specify a summary function and a field to be summarized. The summary functions allow you to count, sum, and average, as well as compute the variance, standard deviation, minimum, and maximum.
- C. Group the data by field values. For example, you can easily obtain the total sales volume for each salesperson in the organization (data grouped by salesperson), the total sales to each company (grouped by company name), and the total sales volume by sales personnel to individual companies (grouped both by salesperson and by company name).
- D. Specify ranges and group values by subranges.

These features are deceptively simple, since their use in combination can produce real insights into the nature of your data. I find that many of the processes that I used to perform with larger, more powerful and harder to use software, can be easily performed with Reflex. For example, as an instructor, I like to see what the distribution of grades are in a class. The Crosstab feature allows me to do this quite easily. I just specify that I want a count of grades in each subrange, where the subranges are given in groups of 10, going from 1 to 100 (0 to 9, 10 to 19, 20 to 29, etc.).

7. You can read data from a variety of files found in many popular data management systems. Reflex can read and convert dBASE, PFS, 1-2-3, Symphony, and ASCII files. The conversion is automatic, but you can intervene to make any changes that you desire.
8. The graphs that you display on the screen can be printed or plotted on a wide variety of devices. As a result, you can use Reflex to extend the capabilities of systems like dBASE II and dBASE III.

continued on page 91

S100 USERS !!!

THUNDER/PLUS IS HERE 8087 ONBOARD

We at LOMAS DATA PRODUCTS never rest on previous success. The THUNDER 186 has been an overwhelming success over the past two and a half years but we just made it better. THUNDER/PLUS is now the most powerful sixteen bit single board computer on the S100 bus. We took all the features of the THUNDER 186 and added more memory and math coprocessor support and just for extra measure we increased the clock speed from 8Mhz to 10Mhz. This has been added to these additional features:

- 2 RS232 serial ports at 19.2 Kbaud max
- 1 Centronics compatible parallel port
- On board floppy disk controller will control both 8" and 5 1/4" disk drives including the new high density PC-AT disk drives
- 512K bytes of onboard memory standard, 1 full Megabyte of memory optional
- Up to 64K bytes of onboard EPROM
- Fully complies with the IEEE 696-1983 specification for the S100 bus
- Optional 8087 math coprocessor at 10Mhz
- Comes with CONCURRENT-DOS, the multitask-

ing operating system that is MS-DOS compatible

- Will directly boot PC-DOS**
- One year warrantee on parts and labor

**PC-DOS may be booted and used to run programs that can be configured to run on terminals or some programs that restrict their operation to ROM BIOS calls for console I/O. Programs that directly access the video RAM will only operate satisfactorily when used in conjunction with our PC compatible color graphics board, COLOR MAGIC.

THUNDER/PLUS is ideally suited for any high performance 16 bit application, including but not limited to: high performance CAD systems, Scientific processing, data base applications, and process control applications. The 8087 coprocessor allows use of THUNDER/PLUS in applications not able to be run on previous 16 bit single board computers.

PRICE:

512K version.....	\$1195.00
1 Megabyte version	\$1395.00
8087 option.....	CALL

IBM COMPATIBLE S100 BUS COLOR GRAPHICS

LOMAS DATA PRODUCTS presents COLOR MAGIC, the most complete compatibility solution for S100 bus computer products. COLOR MAGIC includes three major hardware sub-functions which allow it to emulate the IBM-PC: An entirely compatible video function, an IBM-PC keyboard interface and an IBM-PC compatible timer interface including IBM-PC sound compatibility. HOW COMPATIBLE IS IT? Currently we are running MICROSOFT's FLIGHT SIMULATOR recognized as one of the severest tests of compatibility. We can also directly boot PC-DOS for the IBM-PC with no alterations. Other programs which have been tested and function without problems are: LOTUS 1-2-3, DBASE III, WORDSTAR and VOLKSWRITER. COLOR MAGIC (16K byte version) .. \$595.00

■THUNDER 186 single board computer provides a high performance 16 bit computer all on one board. It is an ideal companion to the COLOR MAGIC to provide a low cost, high performance IBM-PC compatible system. The 8Mhz 80186 offers 10Mhz 8086 performance. THUNDER 186 provides all the components necessary to form a complete system including: 256K bytes of no wait-state RAM, 2 serial ports, a parallel printer port, high performance floppy disk controller controls both 5 1/4" and 8" drives simultaneously, full IEEE 696 (S100) bus for system expansion. The COLOR MAGIC

and THUNDER 186 combine to provide performance you won't find in other compatibles. The price includes the powerful Concurrent DOS operating system. THUNDER 186 \$1195.00

■NV-DISK is a solid state memory with software to emulate a disk drive under MS-DOS, Concurrent DOS, and CP/M-86. NV-DISK is entirely COMPUPRO software compatible allowing COMPUPRO users to take advantage of the lower cost and battery protection support offered by NV-DISK. It offers the advantage of high speed access and no moving parts. It can be battery protected to allow data to stay even while powered down. The board is available in either 512K or 2 Megabyte configurations and multiple boards may be used to create disk drives with up to 16 Megabytes of storage.

NV-DISK 512K, \$595.00 2 MBYTE, \$995.00

■MEGARAM is a high density, high performance dynamic RAM board with up to two Megabytes of storage. Megaram offers no wait state performance in 8086 systems with up to 10 Mhz processors at a fraction of the cost of comparable performance static RAM.

MEGARAM 1/4 MBYTE, \$475.00
1/2 MBYTE, \$595.00 1 MBYTE, \$795.00
2 MBYTE, \$1095.00

IBM-PC COMPATIBLE S100-BUS SYSTEM

LOMAS DATA PRODUCTS offers IBM-PC compatible systems with performance far ex-

ceeding that available from IBM. You can purchase systems offering performance of an eight Mhz 8086 or up to the performance of an 8Mhz 80286. Each system is capable of supporting 8Mhz math coprocessors. Our 8Mhz 80286 system offers IBM-PC compatibility while offering up to 2 times the performance of the IBM-PC-AT. For applications where PC compatibility is desirable but higher performance is a benefit or requirement LDP offers the only viable solution.

WHO IS LOMAS DATA PRODUCTS ...

LOMAS DATA PRODUCTS has been shipping 16 bit microprocessor S100 bus products for five years. We have earned a strong reputation for reliability and performance over these five years. We were running MS-DOS (SCP-DOS/PC-DOS) before IBM knew the operating system existed. We offer a wide range of 16 bit operating systems including MS-DOS, CP/M-86, CONCURRENT CP/M-86 and CONCURRENT DOS. All our products are backed by a one year guarantee. We offer no 8 bit products and concentrate entirely on high performance 16 bit systems. If you are looking for the highest performance possible on the S100 bus, you can be sure LOMAS DATA PRODUCTS offers it.

The S100 Bus Experts.

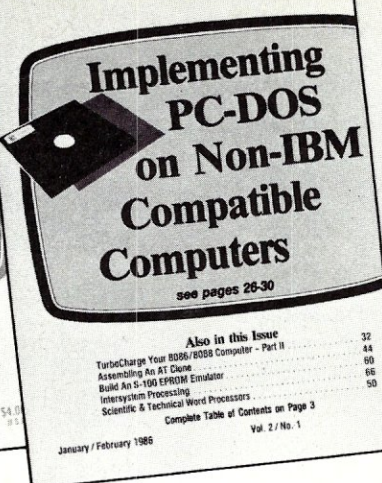
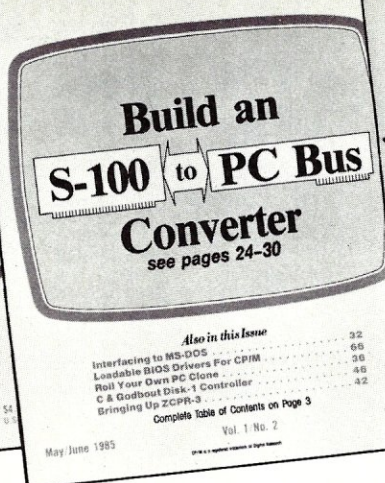
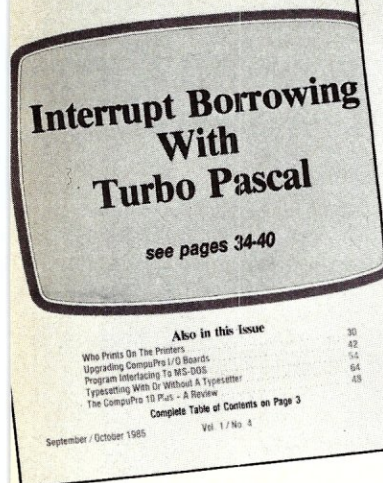
LDP

Dealer inquiries invited.

MS-DOS, trademark of Microsoft
PC-DOS, trademark of IBM
Concurrent CP/M-86, concurrent DOS, trademark of Digital Research

LOMAS DATA PRODUCTS, INC.

182 CEDAR HILL STREET, MARLBORO, MASSACHUSETTS 01752 ☐ TELEPHONE: (617) 460-0333 ☐ TELEX: 4996272



The only magazine by and for advanced micro users.

Micro/Systems Journal is the independent magazine for advanced software and hardware hackers who use MS/DOS, CP/M, MP/M, TurboDOS, Concurrent-DOS and Xenix. An important resource for programmers who use C, Turbo Pascal, Forth, Lisp and Assembler. It also covers hardware based on the S-100, PC and AT busses, as well as powerful single board and multi-user systems. Its name — MICRO/SYSTEMS JOURNAL!

Each issue is packed with practical articles...in depth tutorials, reviews, hints...the latest information on SIG/M, PC/Blue, C-User Group and PC-SIG Public Domain Software. Advanced topics such as 16-bit and 32-bit microprocessors, high performance data acquisition systems, interfacing to peripherals, patching application software packages for improved performance...and lots more! Strictly technical info...no fluff...stuff to keep every hacker up-to-date on the ever-changing micro technology.

Edited by Sol Libes, former editor of Microsystems, former columnist in Byte, PC Tech Journal, PC Week, Computers & Electronics and other magazines and author of 15 books. An authority in the field since the early seventies.

Here are examples of topics covered in recent issues.

- Build an S-100 to PC-Bus Interface
- Loadable Drivers for CP/M
- Structured Programming with Microsoft's M80 Assembler
- Review of Three 16-Bit Lisp & Prolog Packages
- Local Variables in Forth
- Installing ZCPR3
- Build a PC Clone for Under \$1,000
- Interfacing to MS-DOS
- Assembly Language Extensions For MS-Basic
- Reviews of Scientific Word Processors, Concurrent DOS and Turbo Pascal Version 3.0

Save Up To 27% Off Newsstand Price
TAKE ADVANTAGE OF SPECIAL LIMITED TIME OFFER

- [] YES! Start my subscription!
- [] 1 YEAR* \$20.00
- [] 2 YEARS** 35.00
- [] 1 YEAR (Canada & Mexico)* 28.00
- [] 2 YEARS (Canada & Mexico)** .. 52.00
- [] 1 YEAR (other foreign)* 35.00
- [] 2 YEARS (other foreign)** 64.00

* 6 issues ** 12 issues

Checks must be payable in U.S. funds by a U.S. bank

Name: _____

Address: _____

City: _____ State: _____ Zip: _____

make check payable to: MICRO/SYSTEMS JOURNAL
Box 1192
Mountainside, NJ 07092

TRUE MULTI-TASKING!

TASKMASTER is high tech, available now, and it works with virtually all DOS software. Give Lotus, Sidekick, Multimate or most any DOS program the advantages of real multi-tasking. It's simple to use, compatible, bulletproof and most of all, it won't slow you down. That's because TASKMASTER only shares your computer when YOU want it shared. At other times, your visible program runs at full speed, waiting for you to easily switch from program to program at the touch of a key. Compatible with most DOS computers including the IBM PC/XT/AT/Jr. series, you can order TASKMASTER today for only \$69.95 + 5.00 Shipping and Handling, VISA and Mastercard.

ORDER LINE
(206)367-0650

Taskmaster trademark Sunny Hill Software.
Lotus trademark Lotus Development Corp.
Sidekick trademark Borland Intl.
Multimate trademark Ashton Tate.

**Sunny Hill
Software**

13732 Midvale North Suite 206
Seattle, Washington 98133



Coming in the next issue!

- * *Control Systems Made Easy*
- * *Multi-tasking With Turbo Pascal*
- * *SCSI Interfacing*
- * *A First Look at the 80386*
- * *Enhancing CP/M*
- * *Intercepting XIOS Calls*
- * *Using Syslib*
- * *Using SYMDEB with NMI Breakpoints*
- * *Reviews - 9600 Baud Modems, Turbo Modula-2, PC-Pro*
and lots more

Data Base Forum

continued from page 88

Reflex is not going to replace software development systems like dBASE III, Paradox, and R:base 5000 when your users need professional quality information systems requiring multiple files, but it may easily become the standard for the smaller, less professional user-built applications. Paradox has an interface that new users will probably find intimidating and I expect the new dBASE III interface will also intimidate most new users. There is a gap in the novice user market that I

think is nicely met by Reflex. Reflex is not intimidating to novice users and it offers important analytical tools not found in current versions of dBASE and similar products. When novice users find they have outgrown a system like Reflex, they may then find that the interface with products like Paradox and dBASE III are not as intimidating. The investment in Reflex is small enough (\$149.95) that later upgrades to another system could not be considered prohibitive. In addition, if the graphing and analytical features of Reflex are not incorporated into the new system, then

Reflex can still be used to extend the features of the new system. [M]

Nelson Dinerstein is an Associate Professor of Computer Science at Utah State University and an active consultant in database applications on micros. He has a Ph.D. in mathematics from the University of Utah. And, he is the author of 8 books on dBase-II/III, Framework, and R:base 5000 applications.

FREE SOFTWARE FROM THE PUBLIC DOMAIN

User Group Software isn't copyrighted, so no fees to pay! 1000's of CP/M and IBM software programs in .COM and source code to copy yourself! Games, business, utilities! All FREE!

	Rent	Buy
IBMPC-SIG 1-390 Diskettes	410.	850.
IBMPC-BLUE 1-154 Diskettes	175.	435.
SIG/M UG 1-240 Diskettes	155.	650.
CP/M UG 1-92 Diskettes	45.	250.
PICO NET 1-34 Diskettes	25.	100.
KAYPRO UG 1-54 Diskettes	65.	200.
EPSON UG 1-52 Diskettes	65.	200.
COMMODORE CBM 1-28 Diskettes	25.	65.

Public Domain User Group Catalog Disk \$5 pp.
Send your check & specify computer format.
Rental is for 7 days after receipt, 3 days grace to return. Use credit card, no disk deposit.
Shipping, handling & insurance \$9.50 per library.
(619) 941-0925 Orders & Technical (9 to 5)
(619) 727-1015 24 Hr. 3 Min. Info Recording
Have your credit card ready!

National Public Domain Software



1533 Avohill Dr.
Vista, CA 92084



BBB

1-800-621-5640 wait
for tone dial 782542



Dysan
CORPORATION

	30 Qty	100 Qty
5 1/4 SS DD 104/1D	2.01	1.86
DS DD 104/2D	2.69	2.49
8 SS DD 3740/1	2.77	2.56
DS DD 3740/2D	3.51	3.25

maxell

★ Fast
Delivery

	5 1/4	8
SS DD MD1	1.42	1.31
DS DD MD2	1.94	1.80
SS DD FD1	3.04	2.81
DS DD FD2	3.31	3.06

Things

5 1/4 8

Flip 'N File Mini XT	19.35	—
Flip 'N File 50	14.94	24.50
Perfect Data Head Cleaning Kits	18.00	11.95
20 Color Coded Labels	.25	.30

CALL 818-706-8602

★ Credit For USA Direct Dial Call
With Any Disk Order. ★

Disks 'n Things

14755 Ventura Blvd., # 713
Sherman Oaks, CA 91403

Free Price List Available

C CODE FOR THE PC

source code, of course

Concurrent C	\$45
LEX	\$25
YACC & PREP	\$25
Small-C compiler for 8086/88	\$20
tiny-c interpreter & shell	\$20
Xlisp 1.4 & tiny-Prolog	\$20
C Tools	\$15

The Austin Code Works

11100 Leafwood Lane

Austin, Texas 78750-3409

(512) 258-0785

PC-PRO is PC-DOS for CompuPro/Viasyn

- PC DOS Ver 3.1
- PC Graphics Support

for information write

Computer House, Inc.

P.O. Box 709 Woodacre, CA 94973

(415) 453-0865

Trademarks: Viasyn Corp.,
PC-PRO - Computer House.

S-100 BARE BOARDS

8086/8087 CPU - plus
2764 or 27128, 8253, 8259

8088 Auxiliary Processor
I/O mapped, 4K EPROM,
4K RAM, prototype area

\$45.00

Each

Call or write for brochure.

Terms: Check or money order only. CA
residents add sales tax. Prices include
UPS shipping.

Applied Innovations

3000 Scott Blvd. Suite 106

Santa Clara, CA 95054

(408) 748-1875

d/MULTI MULTIUSER dBASE for TurboDOS

TRUE File and Record Locking as easy as
d-BASE-II. Unlimited users can perform the
magic of dBASE in the program or
interactive mode

- * TurboDOS 1.3 or 1.4
- * No Peeks or Pokes
- * System Date and Time Functions
- * Printspooler Controls up to 16 printers

Martian Technologies . . .

**.CREATIng Multi-users from
Single-users around the world**

CALL FOR DETAILS

Martian Technologies

8348 Center Dr., Ste.-F, La Mesa, CA 92041
(619) 464-2924



MATRIX CALCULATOR v 1.20

- 70+ built-in matrix operations incl.
 - Inversion, Sys of Eqns, Eigenanalysis
 - LP-Simplex, Sensitivity, Param. Prog.
 - Multilinear Regression and Statistics
 - Numerical Integration and Dif. Eqns
 - Newton-Raphson method for nonlin Eqns
 - Jacobian, Hessian, 1-d Optimization
- Very Flexible and User Programmable
- Programmable Editor/Printer can make complex matrix in one command!
- \$59.95 with new 160 page manual
- Source code in Pascal optional at \$60

SoftTech Inc.

1-313-544-8544

18505 W. 8 Mile

Detroit, Michigan 48219

C Users' Group

Over 90 volumes of public
domain "C" software including:

- compilers
- editors
- text formatters
- communications packages
- many UNIX-like tools

Write or call for more details

The C Users' Group

Post Office Box 97

McPherson, KS 67460

(316) 241-1065



PROGRAMMERS: Now your software can support over 150 video display terminals.

If you write software you need this book. It contains a tutorial on programming for video display terminals plus data sheets to support over 150 VDTs, including **cursor positioning, clear screen, clear-to-end of line, row and column numbering, video attributes, function keys and much more!** You'll find detailed information needed to support a wide range of VDTs immediately, getting your programs to market much earlier.

"Highly recommended for programmers."

— Jerry Pournelle, BYTE Magazine

335 pages, 6" x 9", softcover, ISBN 0-936158-01-8
15-day money-back guarantee. Texas residents add sales tax. Foreign residents specify and add shipping. Send \$30 check, MC or VISA, to:


Atlantis Publishing Corporation
Dept. 205, POB 59467, Dallas, TX 75229

UNIX File

continued from page 68

had an opportunity to examine and so don't feel prepared to comment on. There are also reference cards on many aspects of UNIX from Pitman, Specialized System Consultants, Cscapes and others.

Most of these books are listed in *Books in Print* or its updates, and can be ordered from your local bookseller. Mail order UNIX book dealers include Cucumber Info Systems, 5611 Kraft Ave, Rockwell, MD 70852, phone 301-984-3539, and Inside Information, 77 Harbord Street, Toronto, Ontario Canada M5S 1G4, phone 416-929-3244.

That's all for this month. I welcome electronic mail and letters on these and other topics, especially suggestions for future columns. Please write to me at Box 603, Station F, Toronto, Ontario, Canada M4Y 2L8. If you have UNIX mail access to the uucp network, mail "ihnp4!darwin!ian". I can't always answer immediately, but I will get back to you; electronic mail gets answered first! 

Ian Darwin is director of Research and Development for SoftQuad Inc, a company providing supported troff publishing software for UNIX. He and his wife live in a rural setting, north of Toronto.

PC SPEEDUP

NEW PRODUCT! — PC-Sprint — \$89⁹⁵

Exec-PC is the specialist in low cost PC speed enhancement.

PC-Sprint™ Works with any PC, color or mono. 4.77, 6.67, 7.38mhz. Speed and reset switch. Plug-in installation

Surprise!™ from Maynard Electronics. Run any 8088 PC at 9.54mhz. Change speeds from Keyboard. No expansion slot used. 100% compatible. - \$249.00

V20 8mhz - Replaces 8088 up to 8mhz (6 or more, \$12.76 or \$10.00 with PC-Sprint) - \$19.95

V30 8mhz - Replaces 8086 up to 8mhz (6 or more, \$12.76) - \$19.95

V20/V30 Tech manual - 224 pages, sold with chip order only - \$7.00

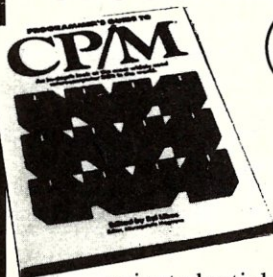
Dealer Inquiries Invited.

All products include chip removal tool, user guide, 90 day warranty, free BBS trial subscription, no extra charge for credit card or shipping. (\$10.00 minimum on charge orders.)

Exec-PC, Inc. P.O. BOX 11268
SHOREWOOD, WI 53211
ORDER HOTLINE RECORDED INFO ORDER BY MODEM
(414) 242-2173 (414) 242-2165 (414) 964-5160



PROGRAMMER'S GUIDE TO



CP/M

EDITED BY SOL LIBES

A collection of all the CP/M oriented articles which appeared in *Microsystems* magazine from Vol. No. 1 (January 1980) up through Vol. 3 No. 1 (February 1982)....an absolute wealth of in-depth information on CP/M. Covers topics such as: Structure and Format, Interfacing, Utilities and Enhancements, Assemblers, Applications and lots more. Also includes a CP/M Programmer's Reference Guide. 200 pages. \$12.95 + \$2.00 shipping & handling.

U.S., Canada and Mexico. \$5 for other countries. Send check for full amount payable in U.S. funds by a U.S. bank. We cannot bill or take credit cards.

Send orders to: Micro/Systems Journal, Box 1192, Mountainside NJ 07082.

Programmer's Reference Cards

IBM/GW Basic Card
C Language Card
UNIX Shell Card
8086/8088 Card
WordStar Card
Z80 Card



Concise comprehensive reference cards that save time and avoid cumbersome manuals. Organized to keep train of thought while programming and enhance productivity. All are 8 1/2"x11" laminated plastic cards printed in two colors, both sides.

\$5.95 + \$1 each (U.S., Canada & Mexico; Foreign \$3) Shipping & Handling; ordering 3 or more cards \$.50 each S&H (\$1.50 each Foreign). Send check (drawn on U.S. bank) or postal money order to: Micro/Systems Journal, Box 1192, Mountainside NJ 07092.

Pocket Reference Cards

CP/M Programmer's Reference Card - \$2, includes postage (U.S., Canada & Mexico; \$3 foreign)

WordStar Reference Card - \$1, includes postage (Foreign \$2)

The following back issues are available at \$4.50 per copy (\$6 foreign, cash, Postal Order, or U.S. bank check) including shipping. If ordering 3-9 copies deduct 10%, 10 or more copies deduct 15%. Orders can be combined with back issues of Microsystems for discount. Send to M/SJ Box 1192, Mountainside, NJ 07092.

MARCH/APRIL 1985 (Vol. 1, No. 1): Bringing up CP/M Plus, Assembly Language Extensions for MS-Basic, New Tricks for CP/M2.2, Building an IBM/PC or XT Clone, Extended Single Density Storage, Variable Size Arrays in C; REVIEWS: dBase-III and 16-Bit Lisp & ProLog-Part I.

MAY/JUNE 1985 (Vol. 1, No. 2): Build an S-100 to PC-Bus Converter, Interfacing to MS-DOS Part-1, Loadable Drivers for CP/M2.2, Roll Your Own PC-Clone, Bringing up ZCPR-3, C & Godbout Disk-1 Controller, Writing Translation Programs in C and Turbo Pascal; REVIEWS: 16-Bit Lisp & Prolog-Part II.

JULY/AUGUST 1985 (Vol. 1, No. 3): Structured Programming With Microsoft M80 Assembler, Local Variables in Forth, Interfacing to MS-DOS Part-II, Data Translation with Turbo Pascal, Implementing Sets with Bit Operations in C, A Unix Mail List System; REVIEWS: Scientific & Technical Word Processors-Part I, Macrotech MI-286 S-100 CPU Card, Slicer System, Concurrent PC-DOS, Coherent Operating System.

SEPTEMBER/OCTOBER 1985 (Vol. 1, No. 4): Who Prints on Printer with CCP/M, Interrupt Borrowing with Turbo Pascal, Upgrading CompuPro I/O Boards, Program Interfacing to MS-DOS Part-III, SIG/M & PC/Blue Public Domain Software Distribution Points, TurboDos INSTALL program; REVIEWS: CompuPro 10+, TEX.

NOVEMBER/DECEMBER 1985 (Vol. 1, No. 5): TurboCharge Your 8086/8088 Computer, Faster Floating Point Math with C, Bringing up CP/M-86K, Program Interfacing To MS-DOS Part-IV, Advanced Machine Level Interface Techniques For Turbo Pascal, Context-Independent Macros for C, Build Your Own PC Program Library Using Public Domain Software, dBase-II Speed Techniques-Part I, Tuning Unix Program; REVIEWS: Peak 68K8-CP, Eureka For CP/M-80, Scientific Workstations.

JANUARY/FEBRUARY 1986 (Vol. 2, NO. 1): Implement PC-DOS on Non-IBM Compatible Computers, Part-I; TurboCharge Your 8086/8088 Computer, Part-II; Assembling An AT Clone; Build An S-100 EPROM Emulator, Transferring Files Between CP/M and MS-DOS Systems; Program Interfacing to MS-DOS Part V; dBase-II Speed Techniques; Unix Public Domain Software; REVIEWS: Scientific Word Processors-Part II; Turbo-DOS PC; Turbo Pascal Support & Enhancement.

MARCH/APRIL 1986 (Vol. 2, No. 2): Implementing PC-DOS On Non-IBM Compatible System Part-II (Conclusion), Program Interfacing To MS-DOS Part-VI, Rolling Your Own PC/XT/AT Clone, Converting From CP/M To MS-DOS On The PC, A C Subroutine For Parsing Command-Line Arguments, Arithmetic Aberrations with dBase-III and the dBase Developer's Release, The Marriage of TEX and Postscript; REVIEWS: C Interpreters, C Source-Level Debuggers, Lomas Data Products' S-100 PC, CompuPro S-100 PC Video Board.

MAY/JUNE 1986 (Vol 2., No. 3): Build An S-100 HD64180 CPU Card, Configuring MS-DOS, Writing MS-DOS Device Drivers (Part 2), Building An AT Clone, Adding A RAM Disk To A CP/M System, Obfuscated C Code Contest, Sorting With Turbo Pascal, More dBase Speed Techniques, Using an RCPM System; REVIEWS: Concurrent DOS, MEX-PC, ConIX, FirstTime.

Microsystems Back Issues

Some issues of the old Microsystems magazine are still available in very limited quantities. They are \$4 per copy (5.50 foreign, cash, Postal Order, or U.S. bank check) including shipping. If ordering 3-9 copies deduct 10%, 10 or more deduct 15%. Orders can be combined with M/SJ back issues for discount. Send to: MS/J, Box 1192, Mountainside, NJ 07092.

1984

MARCH: MS-DOS Overview Part-I, Enhancing MP/M-II Part-2: adding login, date and time functions, Julian date Conversions, I/O changes for Fortran-80, CP/M Mass Renaming by Filetype, CP/M BIOS public domain enhancements, Power Failure Backup for S-100 systems; REVIEWS: MS-DOS for CompuPro From Computer - House, Lomas Lighting One, Dual Systems SIO4, MagicBind, SAL/80K.

JANUARY: Enhancing MP/M-Part 1; Installing MP/M; Add Currency to MP/M; Two Users on CP/M; Relocating Assemblers & Linkage Editors-Part 3; S-100 Wait States; REVIEWS: MP/M-8/16, ProComp-8, Paragraphics Game Board, ProLog.

1983

NOVEMBER: Intro to 80286, 68000, and 16032 Microprocessors; Intro to Local Area Networks-Part 2; Extended Memory Management for older S-100 Systems; Notes on Microsoft Fortran-80; Building S-100 Parallel Ports; REVIEWS: CompuPro CPU-68K, System 8/16, Xenith Z-100, Nevada & Ellis Computing Fortran.

OCTOBER: Intro to Local Area Networks, Part-1; Build Low-Cost LAN; Build S-100 Bubble Memory Card; Use Radio Shack Model 100 portable with a CP/M system; Write Menu Driven Utility for Setting Printer Options; North Star Improvement; True Z-80 Random Number Function; Hide Code in Basic REM statement; Machine Code loader for MBasic; Increase Single-Density Disk Formatting; Relocating Assembler & Linkage Editors, Part-2; Run MX-80 with North Star; User Group Directory; CP/M-86 Versus CP/M-80; REVIEWS: CP/NET, QBAX, S-Basic.

AUGUST: XERA Program; Logging-On CP/M; WordStar Date/Time Patch; Find Location of Variable in North Star Basic; Prevent System Crashes During Warm Boot; Enhance Spreadsheet Print Files; Plot-

ting Package-Part 3; Run WordStar under TP/M; 50-line Text Formatter; Using the LU Utility; User Areas under CP/M; REVIEWS: Stiff Upper Lisp, MuLisp-80, Supersoft Lisp, Cromenco C-10, Access Manager, Fancy Font, Computime SBC-880 S-100 card.

JULY: Using RCPMs; RCPM Directory; PIP Data Between Computers; Toward Smarter Modem Programs; Interface MX-80 via Parallel Interface; Digital Audio On CP/M System; Customize CP/M CBIOS; Plotting Package Part-2; REVIEWS: DRI PL/I-86 and PL/I-80, S-100 PMMI MM-VTI.

JUNE: Plotting Package Part 1; Drive HP Plotter; Laboratory Graphics Applications; Console Keypressed interrupts; Customize Wordprocessor Keyboard; WordStar Patch for H-19/Z-19 Terminal; Relocateable Code; REVIEWS: Graftalk, Jes S-100 Graphics Controller, ZCPR2.

APRIL: IEEE-488 Tutorial; Interfacing to Lab Instruments; CP/M-86 System in Lab; Implementing CP/M+ PART II; Build Simple S-100 Card Extractor; Macros & MacroAssemblers; REVIEWS: Pickles & Trout S-100 488 Controller; CP/M Utilities; Morrow Decision I.

1982

NOVEMBER/DECEMBER: CP/M vs. MS/DOS; CP/M-86 vs. MS-DOS; Intro to ADA Part 2; Virtual Disk for NorthStar; CP/M Program Auto-execute; Macros & Macro-Assemblers; REVIEWS: Janus, Aztec-C, C/80, Morrow S-100 M26 Hard Disk System, Teleram S-100 Bubble Memory Card, Jade S-100 Bus Probe.

1981

NOVEMBER/DECEMBER: Introduction to the C Language, Virtual Segment Procedures, Little-Ada Part-II, A Disk Alignment Routine, Northstar DOS/BIOS Directory & File Conversion using UCSD Pascal; REVIEWS BDS-C, Small-C, Tiny-C and Whitesmiths-C; Tarbell Double-Density Disk Controller.



THE BOOK MART

The following books can be ordered through Micro/Systems Journal. Prices include shipping (NJ residents add sales tax) with prices shown for U.S., Canada and Mexico. Add \$5

for other countries. Send checks for full amount payable in U.S. funds by a U.S. bank. We

cannot bill or take credit cards. Allow 15-20 days for order to be processed. Sends orders to Micro/Systems Journal, Box 1192, Mountainside NJ 07092.

NEW WORTHWHILE BOOKS

A Unix Primer by Lomuto & Lomuto. A very good introductory book. \$24.95

Unix For The People by P. Brown et al. A good non-technical introduction. \$22.95

Preparing Documents With Unix by C. Brown et al. How to use nroff/troff and the editor. Includes templates and a glossary. \$22.95

The Unix Programming Environment by Kernighan & Pike. Recommended for the programmer. \$20.95

Unix & Xenix: A Step By Step Approach For Micros by Topham & Truong. \$22.95

The Unix C Shell Field Guide by Anderson & Anderson. \$20.95

Unix On The IBM PC by Microtrend Inc. \$15.95

Programmer's Guide To Lisp by Tracton. \$12.95

The Handbook of Microcomputer Interfacing by S. Leibson. \$15.95

Microcomputer Interfacing Handbook: A/D & D/A by J. Carr. \$11.95

TURBO PASCAL Language

Using Turbo Pascal by Steve Wood. For new Turbo Pascal (V3.0) programmer. Contains lots of code examples. \$20.95

Programming With Turbo Pascal by David Carroll. By the author of the Turbo Pascal Corner in this magazine. Includes disk with limited version of Turbo Pascal and lots of sample programs; 310 pgs, \$34.95

80286/8086/8088

The 8086 Book by R Rector & G Alexy. A guide to 8086 and 8088 hardware and software. Includes examples of interfacing and applications. \$19.95

80286 Programmer's Guide by B. Childs & S. Evanczuk. \$15.95

80286 System Guide by B. Childs & S. Evanczuk. \$15.95

C Language

C: A Reference Manual by S Harbison & G Steele. Without doubt the best reference manual for C. Don Libes (C Forum author) uses this as his reference. \$20.95

C Programmer's Handbook by AT&T. A concise reference. \$15.95

C Made Easy by Herbert Schildt. For programmers looking to make transition from Basic to C. \$16.95

The C Library by Kris Jamsa. 125 routines from macros to UNIX utilities. \$15.95

C Programming Language by Kernighan & Richie. \$23.50

Learning To Program In C by Thomas Plum. Considered one of the best books on C language. \$28.95

68000

68000 Assembly Language Programming by G Kane, D Hawkins & L Leventhal. Covers topic in explicit detail and contains many program examples. Also useful as a reference ...very complete. Even good for programmers new to assembler. \$19.95

Self-Guided Tour Through The 68000 by M. Hyman. \$15.95

CP/M

Programmer's Guide To CP/M Edited by Sol Libes. Collection of all CP/M articles from Microsystems magazine (Jan 1980 to February 1982). \$14.95

Those wanting to customize their BIOS and add CP/M enhancements will find following books useful.

CP/M Solutions: Improving CP/M by Barbier. \$15.95

Programmer's CP/M Handbook by Johnson & Laird. \$22.95

Following books, by Dave Cortesi, are recommended for learning more about CP/M.

Inside CP/M \$27.50

Inside CP/M-86 \$18.95

The Osborne/McGraw-Hill CP/M-86 User's Guide Includes Concurrent-DOS, Concurrent CP/M-86, DR/Net, DR-EDIX, and DR/Talk. How to use these packages more effectively and trouble shoot hardware and software problems. \$19.95

CP/M Assembly Language Programming by Lindsay. \$19.95

16-Bit SIG/M Public Domain Software Directory An index, by application, of all 16-bit CP/M-86, MP/M-86 & CP/M-68K in the SIG/M library (almost 200 programs), cross-indexed to volume number so you can locate them easily. \$2

S-100 BOOK BACK IN STOCK

We have located a small number of copies of **Interfacing To S-100/IEEE-696 Microcomputers** by Sol Libes & Mark Garetz. This is the definitive book on the subject. It is no longer being published and when this quantity is exhausted there will be no more. So hurry and send in your check. \$24.95

IBM-PC/XT & COMPATIBLES

Inside IBM PC: Access to Advanced Features & Programming by Peter Norton. Detailed analysis of MS-DOS, PC-system architecture, ROM-BIOS service routines and much more. \$20.95; Companion disk \$65.95

Peter Norton Programmer's Guide to IBM PC by Peter Norton. For intermediate and advanced users. For assembly language programmer. \$20.95

Assembly Language Programming for the IBM PC/AT by Leo Scanlon, 320 pgs, \$23.95

Programming With MS/PC-DOS Interrupts in Assembly Language by Thom Hogan & Roger Chapman, 288 pgs, \$20.95

Debugging Techniques for IBM PC Basic by Jerome Corsi & Williams Hills, 288 pgs, \$20.95

8087 Applications & Programming for the IBM PC, XT, & AT by Richard Startz, 320 pgs, \$24.95

Assembly Language For IBM PC DOS by Joe Dornier. Book \$20.95; Disk \$40.95

PC/BLUE Software Directory Catalog listing contents of public domain software disks in PC/BLUE library. Includes cross-reference index by application and other useful information for PC/MS-DOS public domain software users. \$5

ZCPR3 & Z SYSTEM MANUALS

The ZCPR3 Manual \$20.95

The ZCPR3 Library Manual \$30.95

ZCPR3 & IOPs Manual \$10.95

ZRDOS Programmers's Manual \$9.95

If ordering all four manuals at one time deduct 10%.

OTHER BOOKS OF INTEREST

Designer's Guide to Disk Drives by Teja & Gonnella. \$20.95

Discover Forth by T. Hogan \$15.95

Operating Systems Design: The XINU Approach by Comer. How to write a Unix-like operating system with much of source code included. \$34.95 (hard cover)

Micro/Systems Journal accepts Classified Ads. The charge is \$6/line (3 lines minimum); 40 characters max./line. Three times frequency

\$15/line; six times \$25; non-profit clubs \$2/line. Logos, special type, etc. are extra charge. Check must accompany ad copy. Send to Micro-

/Systems Journal, Box 1192, Mountainside NJ 07092.

\$995 LAP-TOP CP/M MACHINE

NEC 8401A with serial and parallel interfaces, 300b modem built-in, battery powered, and 16 line x 80 col. display. **Interface Technology, Inc.**, Box 3040, Laurel MD 20708, (301)490-3608.

FOR SALE: 2 Mitsubishi 8" DS/DD floppy drives, model M2894-63B-12U, CompuPro DISK-1 Rev 171EDMA floppy controller, Full Documentation, \$500 + shipping. Steve Stolen, 4410 Lehann Dr., Kirby TX 78219; (512)662-7318(eves).

JONTEL BUSINESS SYSTEM

The Jontel Business System is a five journal accounting package based on a dual disk CP/M computer. Included with the seven disk package are mailing and inventory programs with 100+ pages of documentation on disk. 8" & 5" disk formats available. \$69.95. Moonlite Computer, 707 Edge Hill Rd., New Bern, N.C. 28560. Source code available for custom installations, compiled CB80. (919)638-6976.

CP/M-80 LIVES on your PC

CP/Mulator puts a 4Mhz 8-bit CP/M-80 emulator in your IBM-PC for only \$99.

- Execute 8-bit CP/M programs at 4Mhz. Stop waiting around for those slow software emulators at twice the price of this speed demon!
- CP/Mulator makes a fantastic CP/M development environment and allows you to use all your CP/M-80 tools, and the IBM-PC tools too!
- Why write off that expensive package that you bought for CP/M-80 just because you upgrade to an IBM-PC? Apply the \$99 solution and keep it working. No retraining or conversion blues!
- CP/Mulator includes the NEC uP-D70108 processor chip which increases the speed of your IBM-PC by at least 10%. 100% compatible with existing IBM-PC software and 8087 compatible.

PURCHASE SOFTWARE ALONE FOR ONLY \$69!!

Source Information
PO Box 2974
Warminster PA 18974
(215)628-4719

FOR SALE: CompuPro CPU-Z 6MHz \$125, Disk 1A and CP/M-80 \$475, 64K static RAM \$97. ALL equipment is in new condition. James Walker, 1205 North Main, O'Fallon MO 63366.

S-100 USERS!!!

S-100 BOARD BANK buys, trades, sells, consigns ancient, modern S-100 cards, etc. **SELLERS:** send full description, BEST price. **BUYERS:** savings on S-100 cards from only \$49; send \$4.37 or more to subscribe to our whimsical, official irregular organ and ad extraordinaire, **THE STATEMENT**. Mailorder only. Write **S-100 BOARD BANK**, Box 344 Mailorder Dept. M2, Olympia WA 98507. 2/1/86

DISK CONVERSION SERVICE

Over 300 CP/M and MS/DOS (PC-DOS) formats. Originals and copies returned in 48hrs. Reasonable rates. Personal service. For information and list of available formats, write: RH Associates, Box 7035, Silver Spring MD 20907, or call (301)587-6230.

Advertiser Index

Anderson Techno-Products	65
AndraTech	65
Applied Innovations	92
Atlantis Publishing	92
Atlas International	37
Austin Code Works	92
BD Software	25
BG Computer Applications	46
BV Engineering	15
Boston Software Works	11
Canada Remote Systems Ltd	47
Carolina Engineering Laboratories	33
Competitive Edge	41-43
Compu-Magic	19
Computer House	92
CompuPro/Viasyn	1
Controlled Printout Devices	65
Coronado Enterprises	46
Curtis Technical Software	20
C Users' Group	92
C Ware Corporation	71
D & W	64
Data Base Decisions	84
Desktop Publishing	27
Digital Decisions	18
Digital Research Computers	21
Disks 'N Things	92
Earth Computer	7
Echelon	45
Educational MicroComputer Systems ..	65

Electronic Control Technology	24
Exec-PC	93
First Capitol Computer	38
Fulcrum Computer Products	51
Gimpel Software	55 & 56
Hawkeye Graphics	86
Holliston Computer	53
Input/Output Technology	87
Integrand	30
Intelligent Computer Design	74
Intercontinental Micro	C3
James River Software	81
Laboratory Microsystems	17
Lattice	9
Lomas Data Products	89
Macrotech International	C2
Magnum Digital	63
Martian Technologies	92
M Global	40
Microcomputer Systems Consultants ..	66
Microdynamics	46
Micro Methods	47 & 85
Micromint	75
Micro Smith	67
Micro Supply Organization	78 & 79
Micro/Systems Journal Renewal	90
Micro/Systems Journal Books	95
Micro/Systems Journal Back Issues ..	94
Miken Optical	67
Mix Software	5

National Public Domain Software	92
New Generation Systems	86
Night Owl	20
Peak Electronics	59
Performics	12
Personal TEX	85
Poor Person Software	59
Port-A-Soft	50
Public Domain Software	83
QCAD Systems	50
Rapid Systems	13
Rational Systems	82
R & L MicroServices	67
Semidisk Systems	C4
Servo Computer	50
Slicer Computers	59
SofCap	77
Softtech	92
Software Wizardry	31
Southern Pacific	33
Sunny Hill Software	91
Systems Facilities	47
Tee Kay	50
Teletek	2
Thinker's Apprentice	69
Tipz	57
TSF	17
Turbo S.I.X.	76
Viasyn/CompuPro	1
Western Wares	32

NOVELL COMPATIBLE

ONE-STOP ARCnet™ LAN SOLUTIONS

Now you can get all the tools you need to build industry standard **TOKEN-PASSING** networks from one source.

InterContinental Micro.

Our networking packages let you configure any combination of nodes (PCs and compatibles, ATs, XTs, Jrs., Z-100s) to a common Fileserver in just a few minutes with our menu-driven installation program.

Active and passive hubs allow you to integrate up to 255 users on a single network with *increased* efficiency as the network grows.

In addition, our stand-alone ARCnet™ Workstation/Fileserver and complete line of S-100 Bus products (single board computers, slaves and controllers) give you and your customers even more options.

ARCnet is a trademark of Datapoint Corporation.

We're also compatible with other industry-leading ARCnet PC™ and S-100 products, and our PC networking boards include an optional 256K of system RAM.

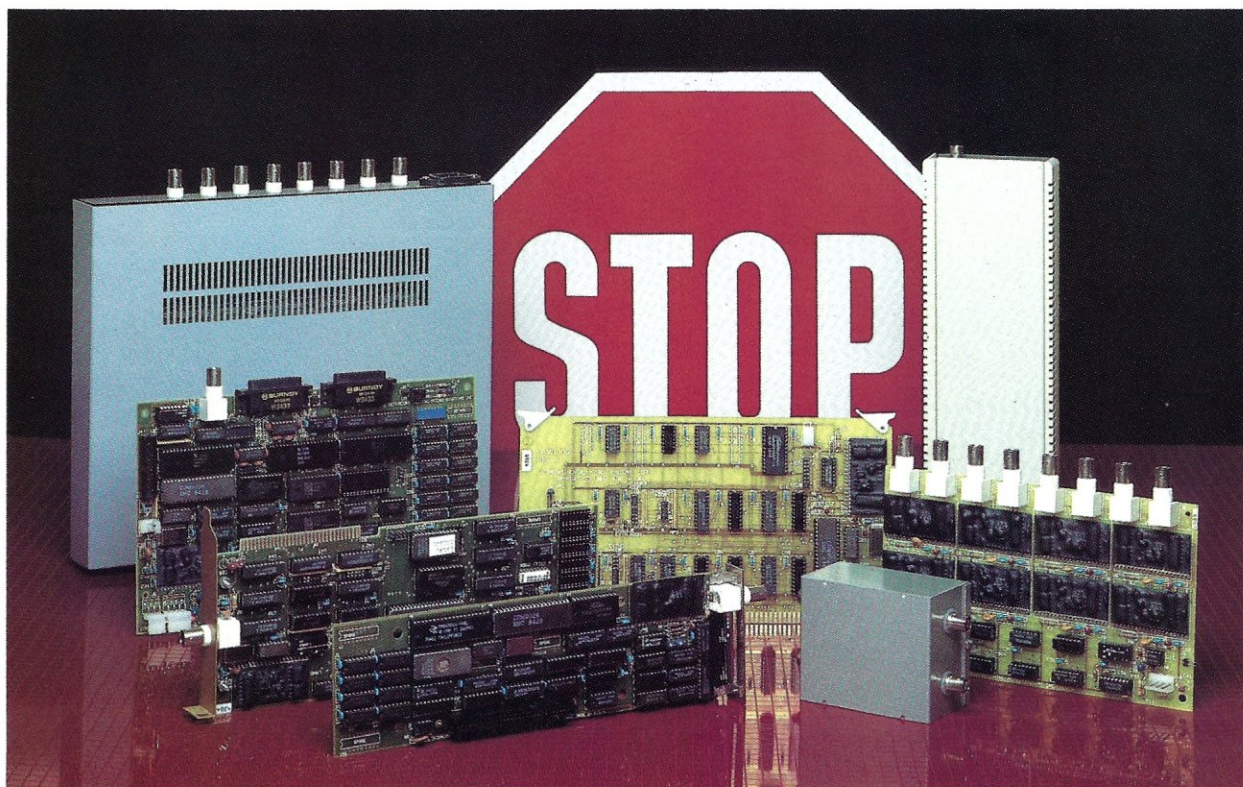
All with InterContinental Micro's reputation for outstanding reliability, exceptional service, and highly competitive pricing.

Call InterContinental Micro today, or circle the bingo number below for ARCnet™ LAN solutions.



4015 Leaverton Court, Anaheim, CA
92807; Phone: (714) 630-0964
Telex: 821375 SUPPORT UD: Easylink: 62562040.

Call for GSA pricing.



The Peak of Performance

SCALE THE HEIGHTS OF PRODUCTIVITY

Sure, you've proven that in your hands a computer is a productive tool. But if you haven't teamed up with a SemiDisk you have heights yet to climb!

IT'S NO MERE RAMDISK

SemiDisk has been leading the way for Disk Emulators since their inception. If you've seen RAMdisks you know what it's like to load programs in an

instant, and read or write files without delay. Unlike alternatives, the SemiDisk offers up to 8 megabytes of instant-access storage while leaving your computer's main memory free for what it does best - computing!

KEEP A GRIP ON DATA

Go ahead, turn off your computer. Take a vacation. With the battery backup option, your valuable data will be there in the morning even if you aren't. You'll sleep better knowing not even a 5 hour blackout will sabotage your files.

NEW LOWER SEMIDISK PRICES THAT WON'T SNOW YOU UNDER

	512K	2Mbyte
IBM PC, XT, AT	\$495	\$995
Epson QX-10	\$595	\$995
S-100, SemiDisk II	\$799	\$1295
S-100, SemiDisk I	\$595	—
TRS-80 II, 12, 16	\$695	\$1295
Battery		
Backup Unit	\$130	\$130

Software drivers available for CPM 80, MS-DOS, ZDOS, TurboDOS, and VALDOCS 2.

SEMIDISK

SemiDisk Systems, Inc.

P.O. Box GG, Beaverton, Oregon 97075

503-626-3104

Call 503-646-5510 for CBBS/NW, and 503-649-8327 for CBBS/Aloha, all SemiDisk equipped computer bulletin boards. (300/1200/2400 baud) SemiDisk, SemiSpool trademarks of SemiDisk Systems.